Commodore
COMPUTER
CLUB
61
L. 5.000
La rivista degli utenti di sistemi Commodore
ESCLUSIVO
Un emulatore
Turbo-Pascal
per il C64
Amiga, trucchi
e consigli
C 16, musica
maestro
L'Amiga
dal dottore
Stampa-sprite
multicolor
E ora le protezioni
a scoppio ritardato
Anno VIII - N. 61 - 25 Febbraio 1989
Sped. Abb. Post. Gr. III/70 - CR - Distr. MePe
systems
INSERTO
CAMPUS

Sommario

RUBRICHE

La copertina e l'immagine del Turbo Pascal sono tratte da Rollergames dalla Qintex Entertainment Inc.

Direttore: Alessandro de Simone - **Caporedattore:** Michele Maggi
Redazione/collaboratori: Paolo Agostini, Davide Ardizzone, Claudio Baiocchi, Angelo Bianchi, Luigi Callegari, Sergio Camici, Umberto Colapicchioni, Maurizio Dell'Abate, Valerio Ferri, Roberto Ferro, Cristina Magnaghi, Giancarlo Mariani, Roberto Marigo, Clizio Merli, Marco Mietta, Marco Miotti, Oscar Moccia, Roberto Morassi, Guido Pagani, Antonio Pastorelli, Sonja Scharrer, Fabio Sorgato, Valentino Spataro, Danilo Toma
Grafica: Arturo Ciaglia, Elena Salvadori
Direzione, pubblicità: via Mosè, 18 - 20090 Opera (MI) - Tel. 02/55500310 - **Redazione:** Tel. 02/5249211
Pubblicità: Milano: Leandro Nencioni (direttore vendite), Angelo Ricupero - Via Mosè, 18 - 20090 Opera (MI) - Tel. 02/55500310
• Emilia Romagna: Spazio E - P.zza Roosevelt, 4 - 40123 Bologna - Tel. 051/236979
• Toscana, Marche, Umbria: Mercurio srl - via Rodari, 9 - San Giovanni Valdarno (Ar) - Tel. 055/947444
• Lazio, Campania: Spazio Nuovo - via P. Foscari, 70 - 00139 Roma - Tel. 06/8109679
Segreteria: Tiziana Sodano - **Abbonamenti:** Liliana Spina (Tel. 02/55500355)
Arretrati e software: Lucia Dominioni (Tel. 02/8467348/9 - viale Famagosta 75 - 20142 Milano)
Tariffe: prezzo per copia L. 5.000. Abbonamento annuo (11 fascicoli) L. 50.000. Estero: il doppio.
Abbonamento cumulativo alle riviste Computer e Commodore Computer Club L. 90.000.
I versamenti vanno indirizzati a: Systems Editoriale Srl mediante assegno bancario o utilizzando il c/c postale n. 37952207
Composizione: Systems Editoriale Srl - **Fotolito:** Systems Editoriale Srl
Stampa: Systems Editoriale/La Litografica Srl - Busto Arsizio (Va)
Registrazioni: Tribunale di Milano n. 370 del 2/10/82 - Direttore Responsabile: Michele Di Pisa
Sped. in abb. post. gr. III - Pubblicità inferiore al 70% - **Distrib.:** MePe - via G. Carcano, 32 - Milano
Periodici Systems: Banca Oggi - Commodore Club (disco) - Commodore Computer Club - Commodore Computer Club (disco produzione tedesca) - Computer - Computer disco - Electronic Mass Media Age -Energy Manager - Hospital Management - MondoRicambi - Nursing '90 - PC Programm (disco) - Personal Computer - Security - Software Club (cassetta ed. italiana) - VR Videoregistrare

ORA CHE CE L'HAI, GUARDA CHE CI FAI

Anche oggi è validissimo lo slogan di una precedente campagna promozionale della Commodore

Recentemente mi è capitato di mettere ordine nella mia affollatissima biblioteca personale.

A soccombere, ovviamente, sono state le "vecchie" pubblicazioni di informatica e mi riferisco, con tale aggettivo, ad un periodo non precedente a cinque anni fa!

Prima di cederle come carta da macero mi son lasciato tentare dalla voglia di sfogliarle un'ultima volta, soprattutto quelle di data meno recente. Inutile dire le risate che mi son fatto leggendo gli articoli e, soprattutto, la pubblicità di quei tempi: "Il nostro calcolatore dispone di ben 16 KRam", affermava un'inserzione; "Disponibile anche il drive per moderni floppy da 5 pollici e 1/4", rincalzava un'altra ditta sulla pagina a fronte.

Anche la Commodore non era da meno. Con le schermate riprodotte dal Vic 20 (e dall'allora nascente C/64) proponeva videogame fantastici e software professionale di notevole pregio; a patto di considerare "fantastica" la primitiva versione di "Space invaders".

A quei tempi, ovviamente, era sufficiente l'uscita a colori per gridare al miracolo; la disponibilità di 3.5 Kram era poi considerata perfettamente normale.

Non sazio della rilettura, in chiave umoristica, delle vecchie pubblicazioni, ho cercato di rintracciare i dischetti contenenti il software relativo a quel periodo; e giù altre risate.

Chi, oggi, acquista o, semplicemente, usa un calcolatore, si abitua rapidamente alle sue potenzialità e stenta a definire eccezionale un moderno package di Desk Top Publishing o un versatile Data base per la cui messa a punto, come invece sappiamo, sono state necessarie schiere di tecnici e mesi di paziente lavoro.

Chi, invece, intuisce il lavoro che si nasconde dietro uno sprite colorato (che, magari, attraversa lo schermo a suon di musica) tende a scoraggiarsi. Il suo pensiero, infatti, corre alle notevoli fatiche compiute di persona nel digitare il programmino, riportato sul manuale di istruzioni del C/64, che muove un piccolo aerostato (piuttosto bruttino) da sinistra a destra, senza scopo; e senza emettere una sola nota.

D'altra parte, chi desidera bruciare le tappe, corre il rischio di sprofondare nella disperazione più nera: segnalazioni di errori incomprensibili, blocco del computer, problemi insolubili si pongono, l'uno dopo l'altro, a chi insiste nel digitare programmi troppo lunghi, "pericolosi" e, comunque, "vietati ai minori" (di alcuni mesi passati davanti alla tastiera).

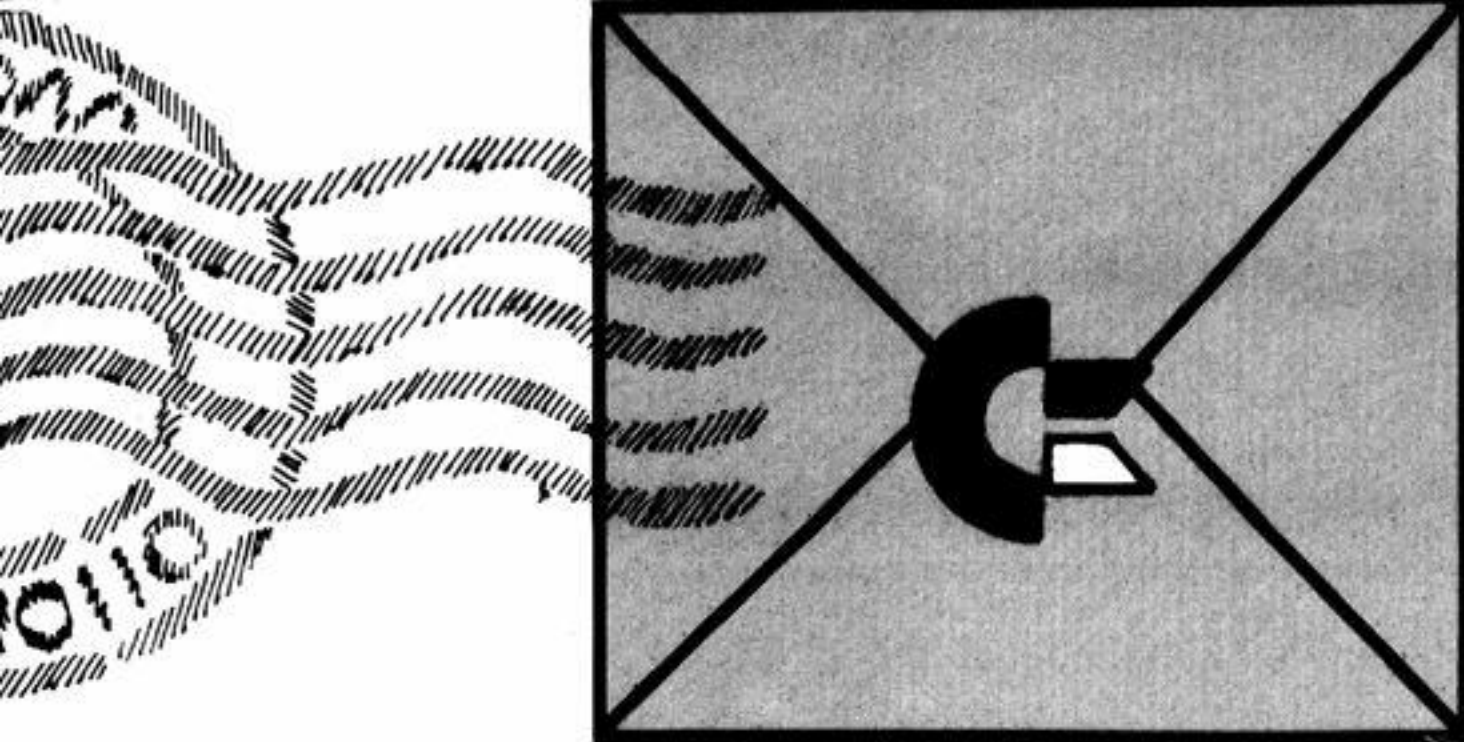
Ma anche a coloro che, di buona volontà ne hanno da vendere, il percorso da compiere, per avere la padronanza completa del mezzo, è lungo e delicato. Come una motocicletta, anche un computer "proibisce" ai neofiti di affrontare curve pericolose con troppa disinvoltura, di superare un dosso limitandosi a sperare che tutto vada per il verso giusto.

In ogni caso, e chiunque si soffermi a ragionare sull'argomento non può negarlo, i grandi traguardi si raggiungono un po' per volta; e sempre partendo dal gradino più basso.

Se, quindi, consideriamo il punto cui è giunto il C/64, non possiamo fare a meno di riflettere su due argomentazioni.

La prima, immediata, è relativa alla straordinaria potenzialità di queste meravigliose macchinette: chi avrebbe potuto prevedere, pochi anni or sono, ciò che il C/64 è, oggi, in grado di offrire?

La seconda considerazione, invece, tende ad effettuare confronti, più o meno validi, più o meno calzanti: se, allora, partendo dai marzianetti siamo arrivati ai favolosi giochi basati sulle manipolazioni del Raster (per non parlare della musica), che cosa ci riserverà l'Amiga che, fin dai primi mesi della sua diffusione, dimostra di offrire caratteristiche più che avveniristiche?

...e il naufragar m'è dolce in questo mare...

la vostra posta

SUBDIRECTORY

□ Per il mio C/64 ho acquistato il drive 1581 con il quale dovrei poter gestire le subdirectory. Che cosa sono?
(Guerrino Stefanelli - Portorecanati)

• L'ottimo drive Commodore 1581 (per il quale, purtroppo, non esiste software, nè esaurienti notizie tecniche riguardo alle sue Rom) consente di memorizzare, ad alta velocità, ben 800 kbyte sulle due facce di un dischetto in formato 3.5 pollici.

La gestione della superficie magnetica è molto sofisticata e consente, tra l'altro, di creare le subdirectory.

Con tale termine si intende la capacità di ripartire il disco in "blocchi logici" che, per l'utente, sono individuabili come più dischetti completamente indipendenti l'uno dall'altro.

L'esigenza delle subdirectory è nata con l'aumento di capacità dei dischetti stessi e grazie all'ottimizzazione delle procedure software. Spieghiamoci meglio: quando, alcuni anni fa, la superficie magnetica dei dischi era relativamente modesta, su ciascun floppy era possibile memorizzare non più di qualche decina di programmi e/o file di vario genere. Sia il progredire della tecnologia, sia l'abilità dei programmatori (che riuscivano a restringere in poco spazio programmi lunghi) hanno consentito la memorizzazione di centinaia di file diversi su ciascun disco. L'introduzione e la diffusione di Hard Disk (dischi rigidi in grado di memorizzare decine di milioni di byte e, quindi, migliaia di programmi) crearono non pochi problemi negli utenti, che erano costretti a rintracciare con enorme difficoltà i file di cui avevano dimenticato il nome.

Per ovviare all'inconveniente, fu escogitato il trucco (perchè di trucco si tratta) delle subdirectory. In pratica l'utente può crearne una dal nome, ad èsempio, "Giochi" e, al suo interno, memorizzare la sua collezione di videogame; sullo stesso disco, poi, può creare la subdirectory "Word processor" all'interno della quale inserire programmi e file per il trattamento dei testi.

Quando, acceso il computer, si richiede la directory principale, lo schermo visualizza i due nomi (Giochi, Word processor) seguiti dal suffisso DIR per ricordare che non si riferiscono a singoli file ma ad una "raccolta" di file (le subdirectory, appunto) i cui nomi, ovviamente, sono "nascosti".

L'utente, a questo punto, può decidere di "entrare" nell'una o nell'altra subdirectory per ricercare ciò che desidera.

E' anche possibile nidificare le subdirectory stesse; ad esempio "Giochi" può contenere, al suo interno, "Arcade" e "Adventure", con chiaro significato sulla suddivisione dei videogame memorizzati.

In conclusione le subdirectory, soprattutto se realizzate su dichi rigidi, consentono di facilitare il raggruppamento e la successiva ricerca di file e programmi.

Per l'uso corretto della partizione, è sufficiente leggere con attenzione il libretto di istruzioni del drive stesso.

6499, COMPATIBILE 128 OPPURE NO?

□ Avete asserito (vedi N.57) che l'adattatore telematico 6499 non è compatibile con il C/128 in modo 64. Su altra rivista del settore (di cui allego fotocopia) si afferma l'esatto contrario. Chi ha ragione?
(Federico Solla - Napoli)

• Purtroppo abbiamo ragione noi, "confortati", tra l'altro, dalle affermazioni di un tecnico della Commodore Italiana.

Alcuni vecchi modelli C/128, per la verità, dimostrano una certa compatibilità con l'adattatore telematico; i modelli nuovi, invece, non funzionano a causa di incompatibilità hardware.

Tale inconveniente, infatti, ha indotto la Commmodore stessa a definire il 6499 come "collegabile al C/64" escudendo (pur se solo impicitamente) la connessione con il C/128.

AVANTI E INDIETRO

□ Quali sono le Poke per far avanzare ed indietreggiare il registratore di un certo numero di giri?
(Carlo Rinaldi - Manfredonia)

• Il registratore è una periferica, del periodo paleo-informatico, di cui esistono in giro alcuni esemplari presso i tombaroli più spregiudicati.

A quei tempi era già un miracolo far partire un motorino elettrico e fermarlo, figurarsi se era possibile farlo girare per un certo tempo e per di più all'indietro! Alcuni Abitanti dei Nastri, resisi conto dell'imminente invasione dei Popoli Delle Regioni dei Drive, tentarono il tutto per tutto per restare al passo coi tempi e cercarono di sviluppare software in cui la magia prendeva spesso il sopravvento sulla tecnica (speed tape, turbo tape, super load, bamboline con spilloni).

Dubbio non c'è veruno che alcuni risultati si ottennero: Anassimene di Commodoria, ad esempio, ricorda nei suoi annali... "Nastrolabio di Registratoria, durante l'assedio dei Popoli dei Drive, realizzò una procedura che consentiva la memorizzazione di una directory all'inizio di ciascun nastro ed una utility che, addirittura, permetteva di far avanzare rapidamente il nastro e di fermare il motorino in corrispondenza del programma desiderato". La macchinosità della procedura, tuttavia, non impediva errori vistosi e la tecnica, benchè funzionante, fu derisa dai nuovi invasori.

Nastrolabio fu condannato a morte mediante impiccagione con il cavo del registratore.

PROBLEMI VARIABILI

□ Quale è la struttura con cui il C/64 memorizza, in cinque byte, un numero in virgola mobile?
(Giancarlo Gotta - Alessandria)
(Antonio Sancina - Calascibetta)

• L'algoritmo è oltremodo complesso ed è stato, in minima parte, affrontato sui due inserti speciali di C.C.C. (N. 43 e 44 "Viaggio nel Basic") che consiglio di rileggere attentamente.

Per iniziare lo studio della codifica, suggerisco di inserire, mediante Poke, diversi valori nei cinque byte di una variabile qualunque (ad esempio: X) e di esaminarne il comportamento, in seguito, con un banale Print X.

FLOPPERIA®

srl

presenta

in esclusiva per l'Italia

la **miglior** cartuccia mai prodotta per Commodore 64 e 128

MK V

Mk V' non solo è la miglior cartridge per effettuare copie di sicurezza del proprio software, ma è anche il più efficace velocizzatore nastro/disco e la più versatile cartuccia di utility esistente.
La sua peculiarità più innovativa è data dal suo **microprocessore interno**, appositamente studiato per sovrapporsi a quello del computer, ed assolutamente invisibile al sistema.
Ad esempio, mentre le altre cartucce si fermano ad un banale "Sprite Killer" per facilitare i giochi, **Mk V' è in grado di trovare automaticamente le "Poke" necessarie per le vite infinite** di qualsiasi programma presente e futuro, senza attendere che siano pubblicate dalle riviste o che qualche smanettone studi il programma. Ora anche voi potete produrre giochi "**trainer**", senza alcuna conoscenza di linguaggio macchina! Inoltre **sprotegge e porta da nastro a disco (e viceversa) qualsiasi programma protetto, anche in multiload** (con i parametri in dotazione); può trasferire molti programmi e files dal formato 5"1/4 al nuovo **1581 da 3"1/2;** velocizza il nastro 5-6 volte oppure 8-10 volte, con velocità selezionabile; velocizza il disk drive come se fosse parallelo (**2 velocità: 202 blocchi in 9 secondi oppure in 6!**), ed è sempre efficace, anche con i programmi che disabilitano i fastload normali. Mk V' incorpora un vero e proprio editor di schermo, per poter cambiare più facilmente e velocemente le scritte nelle schermate o nei programmi; **funziona da interfaccia parallela**, per collegare una qualsiasi stampante standard Centronics al C64/128 e di usarla all'interno di qualsiasi programma, anche grafico; stampa o salva in qualsiasi momento la schermata o gli sprites di un gioco, per alterarli a piacimento. Aggiunge nuovi comandi al Basic, monitor L/M e disk, crea serie di immagini in sequenza su nastro, e tantissime altre cose ancora.

Per Commodore 64 e 128 (in modo 64), con qualsiasi registratore o disk drive, originali o compatibili.

In offerta a solo £. 115.000

(IVA e spedizione compresa!!!)

Viale Monte Nero, 31
20135 Milano

Tel. (02) 55.18.04.84
Fax (02) 55.18.81.05

AIUTATECI A SERVIRVI MEGLIO

Spesso alcuni lettori, che dichiarano di possedere numeri arretrati del nostro periodico, porgono quesiti le cui risposte sono già state esplicitamente pubblicate (in occasione di risposte ad analoghe domande) oppure sono contenute in articoli presenti nei fascicoli in loro possesso.

Per evitare di ripetere argomenti già trattati, pertanto, ricordate di indicare sempre, nelle lettere che ci inviate, i numeri dei fascicoli in vostro possesso: potremmo infatti indicarvi, se esistono, gli articoli che, in un modo o nell'altro, possono chiarire gli argomenti richiesti.

Si ricorda ai lettori che non ci è possibile rispondere privatamente, nemmeno se si acclude l'affrancatura per la risposta.

Solo in seguito si sostituisce il RTS con JMP EA31 e si provvede a modificare il puntatore 314 / 315.

L'errore che hai commesso, quindi, è quello di aver dirottato il vettore di Interrupt verso una zona in cui non... c'è nulla! La routine che dirotta il vettore (da utilizzare una sola volta) può essere allocata in una parte diversa da quella in cui è memorizzata la routine vera e propria.

Inoltre è bene che la routine sia allocata in una zona "protetta" dall'invasione del Basic. La routine da te scritta si trova proprio nel bel mezzo della Ram dei programmi Basic... Sei sicuro di aver settato correttamente il Top di memoria?

PC SONORO

□ Il mio PC non possiede l'altoparlantino e, di conseguenza, non emette suoni. Sul libretto di istruzioni c'è scritto che è possibile inserire un altoparlante sul connettore J6 (pin 1, 2). Di quale impedenza deve essere l'altoparlante?
(Nicola Bosco - Palermo)

• Di solito un qualsiasi altoparlante è adatto alo scopo; stai attento alle connessioni e non provocare corto-circuiti con la parte metallica dell'altoparlante stesso.

QUALE ACCESSORIO?

□ Vorrei collegare la mia stampante non Commodore (con interfaccia Centronix) al C/64 e vorrei sapere quale interfaccia acquistare per andare sul sicuro. Potete aiutarmi?
(Sandro Forti - Roma)

• Primo: recarsi da un negoziante serio e chiedergli se l'interfaccia che vende (da C/64 a Centronics) risolve effettivamente il problema di collegamento.

Secondo: rassegnarsi a non veder funzionare alcuni programmi (soprattutto grafici) che, con una modesta stampante Commodore Mps-803 e/o 1200 funzionano perfettamente.

INTERFACCIA MISTERIOSA

☐ Possiedo una stampante Commodore MPS-1200 di supporto al mio C/128. Al momento dell'acquisto mi venne garantita la possibilità di sostituzione dell'interfaccia C/128 con una compatibile IBM. Però, venuti al dunque, nessuno mi seppe dare indicazioni precise: non l'Assistenza regionale Commodore (a Genova) che all'epoca, circa un anno fa, non conosceva nemmeno la stampante (per inciso: quasi tre mesi sono passati per recuperare a Milano gli schemi elettrici). Una successiva lettera alla Commodore Italiana mi ha fruttato un bel manuale in italiano, ma dell'interfaccia nemmeno una parola.
(Segio Morana - Genova)

AMERIKANO

☐ **Il mio C/64 (acquistato in America) è privo di alimentatore. Posso sostituirlo con uno italiano?**
(Stefano Tronca - Caroloforte)

• Il mio primo C/64 era un modello americano e ricordo che riuscii ad utilizzarlo grazie ad un secondo alimentatore che, da 220 volt (Enel), portava la tensione a 110 volt, valida per attivare l'alimentatore originale. Non penso che vi siano altre difficoltà, tranne il fatto che hai bisogno di un monitor dal momento che il televisore domestico (italiano, cioè sistema PAL) è incompatibile con quello d'oltreoceano (Standard NTSC).

Dal momento che, però, hai deciso di comprare un Amiga (spero in Italia...) non vedo di che preoccuparsi.

Sui numeri di C.C.C. che possiedi sono stati affrontati, più di una volta, gli argomenti che chiedi nella seconda parte della tua lettera: rileggili con attenzione!

PROBLEMI DI CARTA

☐ **Riportando la schermata grafica Hi-Res del mio C/64 (esempio: schermate Gw-Basic emulator, Simon's Basic e così via) sulla mia stampante Mps-803, il disegno non occupa il foglio per l'intera larghezza. E' possibile ingrandirlo? Quali vantaggi otterrei, in tal senso, acquistando una 803 compatibile più costosa?**
(Rocco Tinnirello - Butera)

• Lo schermo del C/64 in modo Hi-Res è largo 320 pixel (puntini elementari), valore fornito da 40 (colonne) x 8 (larghezza matrice carattere visualizzabile).

Una qualsiasi stampante (perfino la super-economica 803) è in grado di riportare 80 colonne, la cui matrice, a seconda dei modelli, può variare da 6 a 8 (e più) puntini. Il valore minimo relativo alla larghezza orizzontale, quindi, è di 480 dot (nome del puntino elementare di una stampante).

E' quindi inevitabile che, riportando un disegno largo "solo" 320 pixel, una parte del foglio di carta rimanga bianca.

Il problema non si risolve acquistando una stampante più costosa (nessuno è ancora in grado di fare il miracolo della moltiplicazione dei pixel e dei dot). Semmai puoi ottenere una velocità di stampa lievemente maggiore. La qualità, invece, rimane pressochè identica sia perchè qualunque stampante, quando "entra" in grafica, annulla eventuali modalità qualitative precedentemente impostate (come il famoso modo NLQ) sia perchè (chissà poi perchè) anche se sono bidirezionali vengono fatte funzionare come monodirezionali.

L'ALTRA FACCIA DEL 1571

☐ **Con il mio C/128, collegato al drive 1571, vorrei formattare l'altro lato di un dischetto, ma non so come fare.**
(Antonio De Giorgi)
(Alessandro Latina - Floridia)

• La differenza esistente tra un drive 1541 e 1571 consiste nel fatto che il primo ha una sola testina di lettura / scrittura; il secondo, invece, ne possiede due, una per ciascuna faccia.

Quando si formatta un dischetto con il C/128, quindi, la seconda faccia viene automaticamente formattata tant'è vero che, alla fine delle operazioni, la directory visualizza 1328 blocchi liberi.

Il drive 1541, al contrario, non è in grado di formattare la seconda faccia, benchè su questa sia effettivamente presente una superficie magnetica utilizzabile.

Il trucco, tuttavia, è vecchissimo: è sufficiente praticare una tacca sul floppy, simmetrica a quella standard, e procedere alla normale formattazione; l'unica seccatura è quella relativa alla rotazione manuale del floppy per accedere all'altra faccia.

INTERRUPT DISPERATO

☐ **Perchè diavolo accidenti e dannazione non funziona il programmino Im che vi invio e che dovrebbe dirottare un banale Interrupt? Sono al limite della disperazione e vi consiglio di rispondermi, altrimenti...**
(Sampey il disperato)
(Vito Catania)

• Mi affretto a risponderti sperando che tu, nel frattempo, non abbia già spedito il pacchetto contenente la bomba ad orologeria (del resto, grazie ai ritardi postali...).

Ricordo ai nuovi lettori, ad ogni buon conto, che l'argomento legato alla gestione degli Interrupt è stato ripreso più di una volta su Commodore Computer Club. Si consiglia, tra gli altri, la rilettura dell'inserto "Principi di multiprogrammazione" apparso su C.C.C. n 46.

Dunque, vediamo di chiarire:

Nelle locazioni esadecimali 314 e 315 del C/64 è riportato il puntatore (nella forma basso / alto) della routine che, ogni 60-mo di secondo, viene attivata automaticamente dal sistema operativo.

Se vogliamo che, ogni 60-mo di secondo, venga eseguita anche una nostra routine, questa deve terminare con un salto a $EA31, indirizzo normalmente contenuto, appunto, in 314 / 315.

PRIMA, quindi, di modificare il puntatore DEVI scrivere la routine che desideri inserire nel ciclo di Interrupt e fare in modo che termini con JMP EA31.

Di solito, prima di attivare l'intera procedura, si scrive la routine desiderata facendola terminare con RTS in modo da controllare il suo esatto funzionamento mediante banali JSR.

LAMENTELE

LA VOCE DEI LETTORI

☐ Sono molto insoddisfatto dell'assistenza Commodore. Acquistata una stampante Commodore 1526, fui costretto, a causa di un difetto, a portarla presso un centro di riparazione per riaverla dopo circa tre mesi e mezzo. In seguito ho acquistato un Amiga 500 che, dopo appena tre mesi di utilizzo, richiedeva una riparazione a causa di strani comportamenti del drive. Dopo un'attesa di due mesi circa, ed un successivo utilizzo di appena una settimana, il difetto è ricomparso e sono in procinto di recarmi nuovamente presso il centro di assistenza.
(Gabriele Bellussi)

VORREI CHE C.C.C...

□ Io vorrei una rivista che contenesse un dischetto dei programmi pubblicati, senza alcuna protezione, che aiutasse ad utilizzare i migliori programmi commercializzati e/o pubblicati (w/p, d/b, grafica), che aiutasse a conoscere il computer iniziando dalla CPU, dai suoi registri e dal set di istruzioni. Non dovete aver paura di ripetervi: i lettori non si offendono. La tabella delle istruzioni Assembler, ad esempio, fu pubblicata sul n. 40, ma se la ripubblicate fate un piacere a tanta gente che non ricorda dove l'ha messa. Dovreste fornire, inoltre, dettagliate notizie sui microprocessori usati dalla Commodore e pubblicare, di tanto in tanto, gli indici degli articoli già apparsi su C.C.C. (vedi Campus n. 50).

C.C.C. insomma, non è una rivista "usa e getta", ma è una pubblicaziine didattica da conservare e consultare per tutta la vita dei computer ai quali è dedicata, purchè questi siano computer di successo e non sfortunati come il C/16 ed il Plus/4.

(Tebro Tanzini - Empoli)

TIRATE D'ORECCHIE

□ Non conosco quasi nulla, o quasi, di programmi commercializzati per il mio C/64 dal momento che abito in una zona in cui non vi sono negozi specializzati.

Leggendo una rivista del settore, nella quale viene descritto un database che gira in ambiente Geos, si fa cenno ad altri programmi che utilizzano tale s.o. di cui non ho mai sentito parlare.

Detto quanto sopra, mi aspetterei da voi una panoramica, se non di tutto il software, almeno di quello più professionale o di maggiore interesse in quanto non tutti risiedono a Milano o in luoghi più forniti di negozi e club.

(Paolo Diglietti - Alassio)

□ Nonostante una qualche (inevitabile) critica che potrei muovervi, non posso che ringraziarvi per il lavoro svolto nel campo dell'alfabetizzazione informatica. In bocca al lupo!

(Luca Santoni - Prato)

ALTRO CHE DISALLINEAMENTO!

□ Dal momento che il mio drive faceva i capricci, l'ho portato presso un centro riparazioni ove mi hanno comunicato un inesorabile "Disallineamento della testina". L'unica alternativa è stata quella di spendere L. 95 mila per la "riparazione". Dopo poche settimane, però, il difetto si è ripresentato, in modo anche più grave. Con grande coraggio, e molta pazienza, ho svitato le quattro viti del drive per vedere che cosa fosse successo. Per farla breve, il difetto era dovuto esclusivamente al consumo del feltrino della "pinza" che, al momento della chiusura dello sportello, costringe la testina di lettura / scrittura ad aderire alla superficie magnetica del disco. Nel caso in cui il vostro drive risultasse "disallineato" (e solo nel caso i cui abbiate una qualche esperienza di piccola manutenzione hardware), ecco la procedura da seguire per rimettere le cose a posto:

Procurarsi un comune feltrino antirumore (dal ferramenta); scollegare il drive dal computer e dal connettore di alimentazione; aprire il drive e liberare la pinza del blocco testina, alzando il circuito stampato principale, svitando le poche viti che ci sono; verificare lo stato di usura del feltrino rotondo; scollare dalla pinza il feltrino usurato; ritagliare dal feltrino, prima acquistato, un cerchietto della stessa grandezza di quello usurato; sostituirlo e rimontare il tutto. Se il drive, inoltre, risulta rumoroso, versate non più di una goccia di olio per macchine da cucire nei vari ingranaggi (non sulla testina, per carità!); pensare intensamente ad una pernacchia ed inviatela telepaticamente al centro di assistenza.

(Bruno Cattivelli - Roma)

CAD PER C/64?

□ Vi sono programmi, per C/64, che consentano di realizzare giornali (anche di sole 10 paginette) oppure CAD?

(Claudio Massetti - Scandicci)

• Con Geos e, in alternativa, con Newsroom, è possibile fare quanto richiesto a patto di divulgare le copie del "giornale" ed i "disegni computerizzati" solo presso amici e familiari comprensivi; ed a patto, inoltre, di voler passare ore ed ore piacevoli(?) alla tastiera.

E' inutile, infatti, illudersi di usare il C/64 in modo "professionale", anche se il nostro C/64 è super-accessoriato con speed-dos, drive e stampanti veloci.

L'unico modo per fare cose appena appena serie è quello di passare ad Amiga oppure Ms-Dos. Credetemi, non c'è alternativa, altrimenti negli studi professionali userebbero tutti il C/64, che avrebbe il vantaggio di costare la decima parte di un computer di elevato livello.

GRAFICI PER C/64

□ Esistono in commercio programmi per realizzare grafici in alta risoluzione?

(Piero De giorgio - Taranto)

• Dipende ca ciò che si intende per "grafici".

Nel caso, infatti, in cui si desideri disegnare in Hi-Res, sono numerosissimi i pacchetti in grado di soddisfare le più disparate esigenze; a patto di non pretendere di

realizzare grafica computerizzata in un'accezione troppo... vasta.

Se, invece, si vogliono realizzare istogrammi o diagrammi a torta, relativi a vari dati digitati in precedenza, altrettanto numerosi sono i programmi (per di più le "lavagne elettroniche" o Spreadsheet) che, dopo aver elaborato i valori introdotti, sono in grado di visualizzare grafici comparativi di vario tipo.

Se si desidera introdurre funzioni matematiche, da studiare in prossimità dei loro punti "critici", la disponibilità si riduce di molto. Sulla nostra rivista, tuttavia, sono apparsi in molte occasioni (tra cui C.C.C. n. 55 e n. 60) diversi programmi che, sfruttando l'alta risoluzione, sono in grado di realizzare curve prodotte da funzioni matematiche.

L'adattamento alle proprie esigenze, inoltre, è piuttosto semplice per chi mastica un po' di Basic.

DISTRIBUZIONI IRREGOLARI

□ Nella mia città le vostre pubblicazioni arrivano in maniera irregolare. Da che cosa dipende tale inconveniente?
(Pietro Maglia - Messina)

• Purtroppo non da noi, ma dalla organizzazione distributiva che, in (troppe) occasioni suscita le lamentele di numerosi lettori.

In casi come questi, purtroppo, non c'è altra soluzione se non l'abbonamento che, a parte il prezzo vantaggioso, riesce ad eliminare i "buchi" della macchinosa macchina organizzativa.

DONNE E COMPUTER

□ Vorrei dire la mia opinione circa le foto definite "sconvenienti" da alcuni lettori. Innanzitutto ritengo che sia molto difficile, per una rivista a larga diffusione, accontentare tutti, ma lamentarsi per le fotografie da copertina è come cercare il pelo nell'uovo. Non sono quindi d'accordo con quei signori che hanno trovato le vostre foto "erotiche e conturbanti" in primo luogo perchè non le ritengo tali e, in secondo luogo, perchè mettere una bella ragazza accanto a quelle belle, ma fredde e quasi magiche, macchine è stata, a mio parere, una soluzione geniale per dare all'insieme un po' di "grazia". A me, in definitiva, fa piacere vedere una bella ragazza sulla copertina, altri ne rimangono sconvolti, altri ancora le guardano così da vicino da notare perfino il segno lasciato dall'elastico del collant (La posta, C.C.C. n. 45). Continuate così.
(Sergio Maschieni - Firenze)

□ Vorrei fare una contestazione pura e semplice senza addentrarmi in pericolosi giudizi "etici". In un sistema sociale in cui lo Stato stesso, con le nostre tasse, sovvenziona i cinema a luci rosse; in cui le televisioni (pubbliche e private) pubblicizzano (esplicitamente ed implicitamente) film violenti a sfondo sessuale e propongono perfino videocassette hard con la scusa di occuparsi di problemi sociali; con le edicole che traboccano di riviste e fumetti porno; con rotocalchi e settimanali di "opinione" che, vantando pretese "culturali" ed "artistiche", pubblicano con insistenza (quella sì, ossessiva) immagini abbondantemente sconce. Ebbene, in un sistema sociale di questo tipo si pretende di vincere l'immoralità cercando di impedire, a C.C.C, la pubblicazione di foto di belle ragazze?

Questo fa veramente sorridere e sorge spontanea una domanda: perchè mettere tanto impegno per togliere una pagliuzza dai nostri occhi lasciando a qualcun altro il compito di togliere una grossa e pesante trave?

Io personalmente non mi sento affatto turbato dall'immagine di una bella ragazza; se qualcuno ha la libido facile, si controlli!

Del resto C.C.C. non ha mai superato i limiti del buon gusto; semmai ci ha regalato, di tanto in tanto, momenti di bellezza e di grazia.
(Federico Cimmaruta - Pisa)

PIRATERIA

□ Gent. Sig. de Simone, sono un suo grande ammiratore e, pur avendo solo 14 anni, non mi interessano i videogame, ma preferisco impegnarmi in qualcosa di più serio. Grazie alla vostra rivista, ed alle numerose prove ed eperimenti effettuati con il mio C/64, posso vantarmi di conoscere (o meglio, di sapere come usare) il calcolatore. Ho creato molti programmi, forse stupidi, ma che mi sono stati utilissimi per approfondire le conoscenze del Basic. Un argomento che richiama il mio interesse è la pirateria, definito gravissimo ed attuale problema nel campo dell'informatica; devo però riconoscere che, per un ragazzo come me, la pirateria ha i suoi vantaggi: oggi non potrei vantare una ricca biblioteca di programmi se non fossero stati possibili i numerosi scambi che ho effettuato con altri appassionati. D'altronde, come potrei acquistare software originale dal momento che ogni programma costa (almeno) L. 15000?
(Eugenio Gori)

□ Vi scrivo unicamente per complimentarvi con voi, perchè non fate schifo (come afferma quel tale nella rubrica "la posta" del n.57 "Strani accadimenti"). Tre anni fa, periodo in cui non sapevo cosa fosse un Print, comprai il mio primo numero di C.C.C. Oggi scrivo routine in linguaggio macchina, e tutto questo grazie a voi che, di mese in mese, superate voi stessi in meglio. Non lasciatevi sconfortare per le opinioni di quell'anonimo lettore: sono molti quelli che, come me, devono la loro preparazione informatica alla lettura di C.C.C.
(Lazzaro Malpoli)

LA VOCE DEI LETTORI

CRITICHE

☐ A mio parere, la parte più interessante ed utile (senza nulla togliere alla validità dei vostri articoli) è la PUBBLICITA', naturalmente specifica del settore. Vi consiglio di aumentare i vostri inserzionisti incoraggiandoli, magari, con una politica di prezzi adeguata. Dovreste, inoltre, commentare, giudicare ed illustrare, periodicamente, i prodotti pubblicizzati. Evitate le paginette destinate al C/16 ed al Plus/4 e parlate dei Commodore NUOVI perchè, in tutta franchezza, chi rimane ancorato al C/16 non ha interesse a comprare riviste tecniche, anche se lo prendono in considerazione. Inoltre gli annunci economici sono utilissimi (ampliate lo spazio ad essi dedicato!); sarebbe opportuno, mese per mese, illustrare dettagliatamente il contenuto delle altre vostre pubblicazioni, almeno per ciò che riguarda il software Commodore compatibile.
(Alessandro Spagnolo - Pomezia)

PER L'ULTIMA VOLTA

☐ Prima di "passare" all'ambiente Ms-Dos vero e proprio vorrei cimentarmi, con il mio C/128 dotato di 1570, con un emulatore che consenta di far girare i pacchetti Ms-Dos compatibili.

(Gaetano V. - Foggia)
(Carmelo C.)

• Quante volte dovrò ripetere che con un C/128 non è possibile usare i pacchetti Ms-Dos? Quante volte dovrò sottolineare che il nostro emulatore Gw-Basic (insieme all'altra nostra utility IBM Reader) è in grado di leggere ESCLUSIVAMENTE i programmi scritti in Gw-Basic e nessun altro (nemmeno i file batch) e, per di più, solo se in possesso di drive 1571? Speriamo che questa sia l'ultima.

COBOL E NUOVI LINGUAGGI

☐ Avete asserito (C.C.C. n. 45) che il Cobol è "un animale informatico in via di estinzione". Nella mia scuola (ragionieri programmatori) affermano, invece, che il linguaggio è indispensabile per gestire grandi masse di dati. Chi ha ragione?
(Giovanni Caglioni - Azzano)

• Vi sono due aspetti da chiarire per meglio affrontare l'argomento. Il primo riguarda la fascia di utilizzatori di un personal computer, alla quale ci rivolgiamo e a cui non può interessare un discorso Cobol in quanto le macchine che ognuno di noi possiede non hanno la struttura (nè la necessità) di elaborare un certo tipo di dati.

Per un home o un personal computer, quindi, vi sono linguaggi decisamente evoluti, in grado di reggere ampiamente il confronto con ciò che può offrire il vetusto Cobol.

Da un punto di vista meno informatico (ma più "commerciale") c'è la necessità, da parte delle grandi aziende di informatica, di rispettare la compatibilità con software già sviluppato negli anni precedenti e di sfruttare al massimo sia le risorse di macchine strutturate in modo molto particolare, sia il Know How (in una parola, l'esperienza acquisita) degli addetti ai lavori che consente loro di realizzare con efficienza determinate procedure.

Non è semplice, in altre parole, radunare i programmatori di una Banca o di una grossa Società e dire:

-Ragazzi, da domani i nostri calcolatori lavoreranno in "C" perchè sembra più veloce del Cobol. Quindi datevi da fare per convertire tutti i vostri programmi finora realizzati-.

A volte, purtroppo, è più economico mantenere in vita brontosauri, che ben si conoscono, anzichè eliminarli per portare a spasso un cagnolino poco conosciuto.

Ne sa qualcosa la stessa IBM (e la Microsoft) che, al solo scopo di consentire la compatibilità con il software precedentemente realizzato (e venduto), è costretta, tra le altre arcaicità, a limitare i nomi dei file ad appena otto caratteri!

AVVISO

Con le risposte che compaiono in questo numero risulta evasa l'intera corrispondenza giunta in Redazione entro il 20 gennaio, compresa l'enorme "eccedenza" verificatasi nel periodo natalizio. Ai lettori, che non avessero trovato risposta ai loro quesiti, si suggerisce di esaminare con cura i fascicoli pubblicati a partire dalla data dell'invio della propria lettera. E' molto probabile, infatti, che, nel frattempo, siano apparsi articoli i quali, magari implicitamente, rispondevano alla domanda; è altresì probabile che la risposta sia rintracciabile nelle colonne della "Posta" dedicate ad altri lettori. Sono state trascurate, infine, tutte le lettere che ponevano quesiti relativi ad argomenti chiaramente ed ampiamente descritti nei libretti di istruzione dei computer Commodore.

risposte rapide

NON SENZA DISCO
(Claudio Massafra - Bari)
(Fabio Bruschi - Milano)
(Francesco Passerini - Perugia)

Non possiamo prendere in considerazione articoli e programmi se questi giungono in Redazione su carta o su nastro. E' assolutamente indispensabile inviarli su disco. Non possiamo prendere, inoltre, in considerazione i progetti relativi a (piccole) realizzazioni hardware in assenza del prototipo.

DISTANZA DI SICUREZZA
(Ivano Tortolini - Piombino)

Un drive può essere collocato anche molto vicino ad un monitor o TV; l'importante è inserire uno schermo protettivo, tra i due apparecchi, realizzato con un semplice foglio di alluminio (del tipo Domopak per conservare i cibi in frigo). I malfunzionamenti del drive, che funziona benissimo se lontano dal monitor, dipendono dal campo elettromagnetico generato dal TV.

TOTOCALCIO
(Alcuni lettori)

Lo speciale "Totocalcio", comparso di recente in edicola, è semplicemente la ristampa del precedente "Speciale Totocalcio", solo che stavolta è allegato un dischetto e non il nastro. La pressante richiesta dei nuovi utenti, unita all'esaurimento dei fascicoli arretrati, ci ha indotto all'operazione editoriale accennata.

PIU' FACILE
(Francesco Scarnera - Grottaglie)

Il programma che hai inviato è simpatico ma potrebbe essere decisamente più breve (e più semplice da digitare) ricorrendo a cicli For...Next oppure ad istruzioni Read...Data.

ADATTATORE TELEMATICO
(Roberto Gaeta - Brindisi)

Con il 6499 puoi collegarti, via telefono, alle banche dati, tra cui le Pagine Gialle e Videotel. Attento alla bolletta del telefono!

SENZA LUMINOSITA'
(Carmelo Cipolla)

Il comando "Color" del C/16 consente di impostare tre parametri relativi, rispettivamente, all'area da colorare, al colore scelto ed alla sua luminosità.

Sul C/128 la sintassi è limitata ai soli primi due parametri: non è possibile regolare l'intensità luminosa.

NON C'E'
(Giacomo Buratti - S. Giuliano)

Non abbiamo mai pubblicato lo schema di un modulatore TV.

DE SIMONE CLUB
(Alcuni lettori)

Il Club da me fondato "quasi" segretamente (e di cui avete avuto, chissà come, notizia) è aperto ad una ristrettissima cerchia di appassionati, dotati di pazienza, esperienza e (tanti) sistemi computerizzati completi. Non posso accettare le richieste di partecipazione, se non dopo un'attenta e rigorosa selezione. Tengo a precisare, tuttavia, che una delle regole del Club impone l'assoluto divieto di trattare argomenti legati ai videogame: si lavora solo con software professionale.

PRINCIPIANTE
(Lorenzo Milicia - Cittanova)
(Mauro Giannotti - Lido di Camaiore)

Se vuoi davvero impegnarti tantissimo per imparare ad usare il tuo C/64, ti consiglio di comprare l'intera collezione di Commodore Computer Club (dai primi numeri ad oggi) e di "digerire" interamente almeno un fascicolo a settimana. Dopo un anno, se avrai mantenuto pazienza e costanza, potrai vantare una cultura veramente mostruosa.

ISTRUZIONE USR
(Valerio Capello - Nuoro)

Nell'inserto Campus del n. 50 (articolo "Affinità elettive tra Basic e lm") è riportata un'esauriente spiegazione dell'istruzione Basic "USR" corredata da numerosi esempi applicativi.

LINGUAGGIO MACCHINA
(Claudio Colecchia - Foggia)
(Gianpaolo Ascione - S.M. La Bruna)
(Vito Catania - Catania)
(Vincenzo Dell'Erba - Bronte)

Dal N. 51 (nell'inserto "Campus") è iniziato un corso sul linguaggio macchina e sull'Assembly del C/64 (a firma di Domenico Pavone). Un fascicolo speciale, comunque, è stato pubblicato tempo fa ("Commodore speciale: linguaggio macchina e routine grafiche di Toma") ed è possibile richiedere anche il dischetto contenente i programmi pubblicati. A proposito: non ti illudere di imparare qualcosa usando il solo registratore a cassette.

C/16
(Tiziano Diamanti - Montevidon)
(Mauro Mingarelli - Velletri)

Ti sconsiglio di dedicare energie al tuo obsoleto computer. Cerca di procurarti, se non un Amiga, almeno un C/64 completo di drive.

AFFEZIONARSI
(Andrea Tranchida - Enna)

La domanda è molto imbarazzante e non so dirti a quale dei miei otto computer sono più affezionato; posso dire che quello che uso più frequentemente è il C/128-D

(per motivi di w/p), ma ognuno di loro mi ricorda momenti particolari (più o meno piacevoli) della mia "carriera".

PLUS/4, A CHI DARLO?

(Stefano Todisco)

Puoi tentare di disfarti del tuo Plus/4 compilando l'apposito tagliando per la pubblicazione delle inserzioni gratuite. Dubito, però, che tu possa ricavare una cifra superiore alle 150 mila lire. Tutti, infatti, sanno bene che il Plus/4 è obsoleto e privo di programmi specifici di un certo interesse.

MONITOR

(Vito Catania - Catania)

Vi sono molti linguaggi interpreti Basic che posseggono, tra gli altri, il comando "Monitor" per lavorare in linguaggio macchina. Per gestire le loro potenzialità (richiamabili con forme sintattiche più o meno uguali per tutti i Monitor) occorre conoscere piuttosto bene il linguaggio macchina e, magari, l'Assembly. A causa della vastità dell'argomento non è possibile rispondere in questa sede.

TURBO PASCAL PER C/64

(Rossella Reali - Brescia)
(Marcello Rauccio - Curti)

La scheda Z-80 per C/64, con la quale può girare il sistema operativo CP/M, è difficilissima da trovare; per di più non esiste il linguaggio "Turbo Pascal" per un C/64 accessoriato con detta scheda. L'Oxford Pascal sembra più adatto allo scopo, ma è un linguaggio un po' vecchio e, per di più, è difficile entrare in possesso del libretto di istruzioni e ne consegue che bisogna usarlo "alla cieca". Se, però, avete un po' di pazienza, vedrete presto in edicola un nostro prodotto su disco che...

HARD COPY IN NEGATIVO

(Un "non maniaco")

Per riportare su carta un'immagine hi-res in negativo è necessario che ogni dot (puntino elementare) "acceso" sia riportato "spento" e viceversa. Non dovrebbe esser difficile introdurre una tale modifica nei programmi che abbiamo già pubblicato, ma non ne vedo una pratica utilità. Tieni conto, comunque, che l'enorme quantità di inchiostro necessaria per stampare un disegno prevalentemente nero, oltre ad esaurire il nastro, provoca deformazioni sul foglio di carta stesso ed obbliga la testina di stampa ad un super lavoro.

SUONI PROLUNGATI

(Massimo Cavalleri - Bussolengo)

Una volta attivato il SID del C/64, questo emette di continuo un suono finchè un'apposita Poke non provvede ad inibirlo. Se il suono persiste fastidiosamente anche con programmi commercializzati (soprattutto videogame) è probabile che vi siano guasti al circuito.

GESTIONALI

(Antonio Antonioli - Massa)

Non disponiamo di software gestionale; dubito, però, che sia possibile trovare in giro un pacchetto per C/128 che sia in grado di svolgere funzioni così complesse: per l'informatica professionale è meglio usare computer Ms-Dos compatibili.

BASIC ALIENO

(Giovanni)
(Lalli Andrea - C. Stazione)

Le istruzioni del programma che ci invii in fotocopia (tratto da un libro della "concorrenza") appartengono ad un linguaggio Basic non originale Commodore; forse è il Simon's Basic, ma non ne sono sicuro. Sarebbe stato meglio se gli autori del libro avessero inserito, nelle REM di intestazione, il linguaggio usato! Il Simon's Basic è da considerare obsoleto nonostante i suoi 114 comandi aggiuntivi descritti nel voluminoso manuale di istruzioni allegato alla confezione.

DOMANDE POCO CHIARE

(Caterina Della Bana)

Purtroppo le domande sono formulate in modo poco chiaro e non è possibile fornire una risposta adeguata.

RESPONSABILITA'

(Carmelo schiavo - Palermo)

Purtroppo non possiamo assumerci la responsabilità di un mancato funzionamento nel caso in cui i programmi riportati sulla rivista non vengono digitati così come appaiono. E' probabile che, digitando il disassemblato mediante un Monitor (invece che con il caricatore Read...Data), sorgano problemi di incompatibilità al momento della compilazione.

MODULA 2

(Raul Luisoni - Novazzano)

Non mi pare che esista la versione "Modula 2" per C/64. Per sviluppare procedure sofisticate, quali quelle consentite da Modula 2, sono necessari computer ben più potenti.

ENCICLOPEDIA

(Luca Vignale - Brandizzo)

Le routine dell'enciclopedia sono destinate ad utenti piuttosto esperti in lm e vanno caricate mediante una procedura più volte descritta nella stessa rubrica.

BIORITMO

(Valter Buccinà - Genova)

Il programma "Bioritmo per C/128", di cui si parla nella lettera, mi sembra troppo macchinoso per proporlo ai nostri lettori, ma è tuttavia interessante. Automatizzando la procedura di inizializzazione, forse, potrei inserirlo in "Directory".

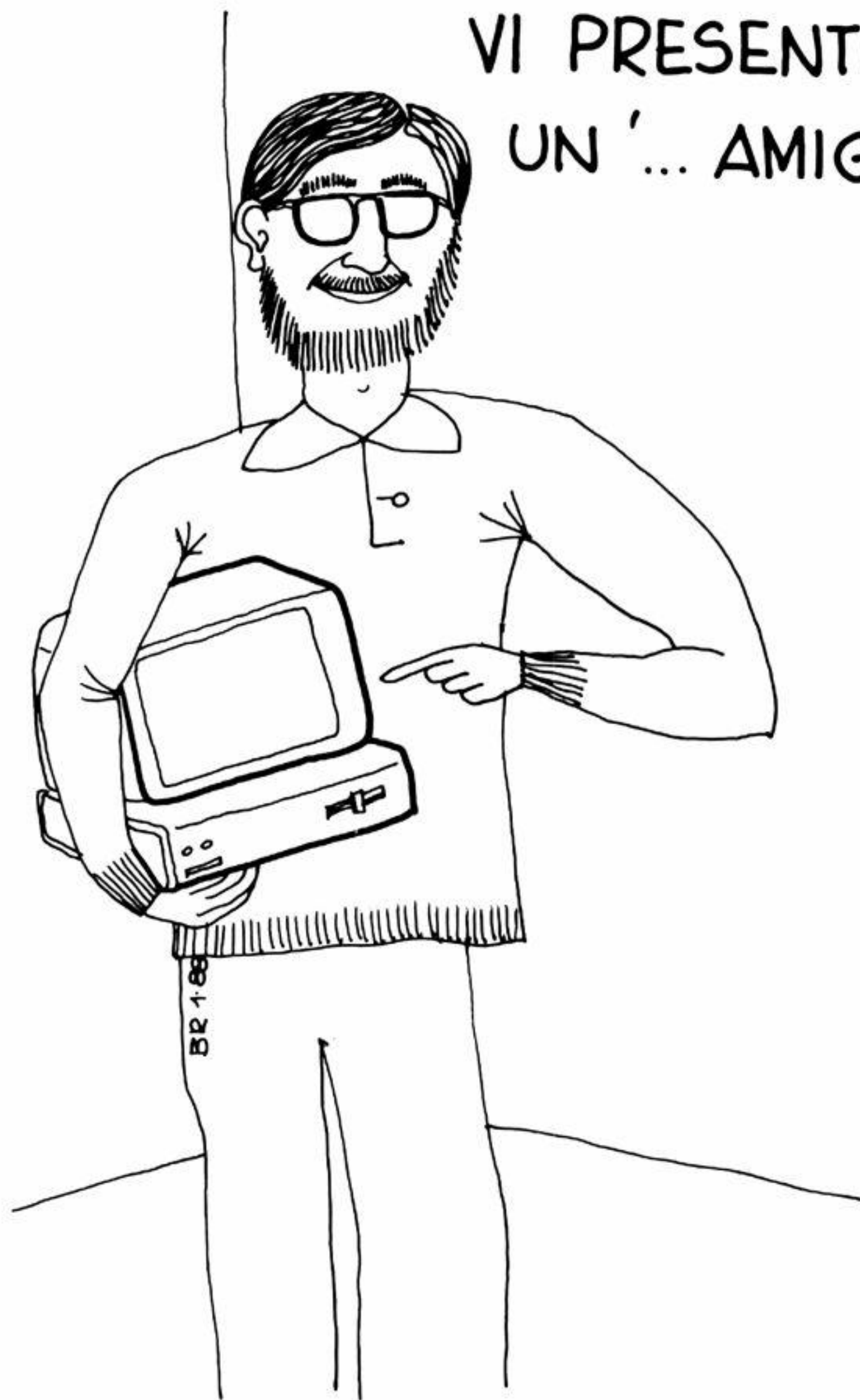
ELEMENTARI E MEDIE

(Leandro Grieco - Villalba)

Non disponiamo di programmi didattici specifici per allievi di scuole elementari e medie.

MIDI PER C/64

(Francesco Salvatori - La Spezia)

L'interfaccia Midi è un dispositivo elettronico che, guidato da opportuno software, è in grado di "pilotare" numerosi strumenti musicali elettronici dotati di analoga interfaccia. Anche per il C/64 vi sono ottime schede di tipo Midi; una di queste è stata descritta ampiamente nel N. 13 della rivista "Personal Computer".

UN COCKTAIL PER AMIGA

Chi non ha dubbi su Amiga scagli il primo mouse

(a cura di Roberto Ferro)

ESPANSIONE DI MEMORIA

Alcuni programmi (ma in genere si tratta di pezzi d'antiquariato di cui esiste quasi sempre una versione più aggiornata) non girano correttamente quando abbiamo un Amiga espanso ad un mega o più. Per ovviare a questo inconveniente bisognerebbe eliminare l'espansione di memoria in modo da avere a disposizione una configurazione con soli 512 KRam. Tale soluzione è piuttosto scomoda e decisamente poco pratica; per questo motivo la Commodore ha pensato di inserire, nel disco del WorkBench fornito insieme alla macchina, una utility di nome "NoFastMem". Lanciando questo programma (che si trova nel cassetto System), il computer si configura come se non fosse installata alcuna espansione di memoria. In questo modo tutti i problemi, causati dai programmi di cui parlavamo, vengono eliminati.

Per riavere il maltolto è sufficiente lanciare una seconda volta la stessa utility NoFastMem ed il gioco è fatto.

Il motivo dei malfunzionamenti dovuti alle espansioni è piuttosto complesso da spiegare. Ne riparleremo, approfonditamente, in uno dei prossimi fascicoli, cogliendo l'occasione per fare chiarezza sulla gestione dell'intera memoria da parte di Amiga.

PROGRAMMI IN BASIC

Quando scriviamo un programma in AmigaBasic, e desideriamo salvarlo, è necessario richiamare l'opzione "Save as" dal menu "Project" oppure usare il comando Save impartito dalla finestra Output. Se vogliamo che il programma venga registrato sullo stesso disco Extras, da cui abbiamo lanciato il Basic, dobbiamo semplicemente scrivere il nome del programma e premere il tasto Return.

Se, invece, vogliamo che sia registrato su un altro disco, allora bisogna usare le convenzioni standard di AmigaDos: indicare il nome del disco, seguito dai due punti e dal nome del file; esempio:
MieiProgrammi:Catalogatore

La procedura vale tanto per Save as da menu quanto per Save impartito come comando; in quest'ultimo caso dobbiamo racchiudere, tra virgolette, il nome del disco e del file. Se desideriamo indicare anche la directory, niente di più semplice: poniamo, nell'ordine, il nome del disco, quello della directory e, separato da uno slash (/), il nome del file.

DISK HAS A READ / WRITE ERROR

Quante volte avete visto comparire quest'odiosa richiesta di sistema?

Quando il drive emette un rumore che fa venire la pelle d'oca, e poi si zittisce facendo comparire un requester simile a questo (ma possono comparire messaggi di diverso tipo, tutti riferiti a difetti vari del disco) significa che gran parte dei dati contenuti nel disco inserito sono in serio pericolo.

La condizione di errore, infatti, si verifica quando Amiga non è più in grado di leggere il contenuto di un disco. Le cause possono essere molte ma, nella maggior parte dei casi, alla base del disastro c'è sempre la bassa qualità dei dischi usati, oppure l'intenso (e pesante) uso che ne fate.

Tutti dovrebbero mettersi in testa, una buona volta, che è meglio spendere qualcosa in più per un disco di marca (sulla cui affidabilità avere un certo margine di sicurezza) piuttosto che risparmire poche lire per vedere il lavoro di lunghi giorni andare in fumo; diffidate, comunque, di chi vuole vendervi dischi ad un prezzo superiore alle 4500 lire, decantando chissà quali meraviglie.

Quando, invece, parliamo di uso "intenso e pesante" ci riferiamo al "togli e metti", magari mentre la luce del drive è accesa, con danno del disco ma soprattutto del disk drive.

Un modo per owiare all'inconveniente della perdita di dati c'è, seppur parziale e non sempre efficace: affidarsi al comando CLI Diskdoctor. Come forse molti già sapranno, il suo utilizzo è di estrema semplicità: entriamo in CLI e digitiamo semplicemente il comando Diskdoctor seguito dal nome del drive. Esempio:

DiskDoctor df1:

Subito saremo cortesemente invitati ad inserire in df1: il disco da controllare (che non deve essere protetto da scrittura) e premere Return. Diskdoctor esaminerà diligentemente ogni byte del disco segnalandoci eventuali anomalie; ci chiederà se può cancellare del tutto i files che contengono dati illeggibili ed infine restituirà il controllo.

Nel caso in cui tra i dati danneggiati vi sia anche il nome del disco, DiskDoctor usa significativamente il nome "Lazarus". Un disco così trattato non è chiaramente molto affidabile per un ulteriore uso, ma dovrebbe esserlo almeno fino a quando non ci siamo preoccupati di trasferire tutti i dati che interessano (e che si sono salvati) su un altro disco.

LA VOCE DEI NUOVI LETTORI

Esaminando la posta che giunge in redazione, capita sovente di leggere quesiti ai quali abbiamo già dato risposta in precedenti numeri di C.C.C. Altre volte si tratta, invece, di domande la cui risposta è contenuta nel manuale fornito insieme all'Amiga. In entrambi i casi le lettere sono lasciate un po' in disparte e se ne rinvia la pubblicazione per fare posto a quesiti di natura più interessante.

Questo mese, tuttavia, abbiamo deciso di accontentare anche chi ha posto domande su argomenti già trattati. Si tratta quindi di nozioni dedicate fondamentalmente ai principianti ed a coloro che hanno acquistato da pochissimo tempo il loro Amiga e sono assetati di qualunque tipo di informazione. Ad uso e consumo di tutti voi, ecco dunque una breve raccolta di suggerimenti, notizie e novità riguardanti Amiga.

Un disco che è andato anche una sola volta in Read / Write Error dovrebbe almeno essere guardato con diffidenza, se non addirittura buttato via.

A volte, però, è sufficiente riformattarlo per utilizzarlo nuovamente.

GURU E OUT OF MEMORY

La comparsa di una Guru Meditation a volte (ma è un caso sempre più raro) è dovuta ad un difetto del programma che stiamo utilizzando, e pertanto non abbiamo molti mezzi a disposizione per difenderci. Molto più spesso, invece, è dovuta alla configurazione dell'Amiga e, in particolar modo, alla memoria. Succede spesso che chi usa una macchina dotata di soli 512 Kbytes si trovi di fronte ad un Guru semplicemente perchè si è esaurita la memoria nel tentativo di vedere in azione le capacità multitasking di Amiga. Se, infatti, interessa utilizzare un programma di gran mole oppure realizzare il multitasking con programmi corposi, è necessario disporre di almeno 1-2 Mega di memoria, altrimenti il rischio è proprio quello di provocare un crash del sistema.

GURU E ANTIGURU

La biblioteca di pubblico dominio (cioè tutti quei programmi che volenterosi programmatori mettono gratuitamente a disposizione di chiunque ne entri in possesso, purchè non tenti di realizzarne un guadagno) spesso offre prodotti di interesse notevole, per i quali saremmo anche disposti a pagare.

Uno di questi è senz'altro il programma GOMF (Get Outta My Face). Si tratta di una breve utility che possiamo lanciare in background (renderla, cioè, attiva senza disturbare il nostro lavoro) e che provvederà a rintracciare le condizioni di errore e intrappolarle prima che si trasformino in rovinose Guru Meditation.

Una volta intercettate, GOMF offre la possibilità di chiudere il task (= programma) che ha causato il disastro per poi restituire Amiga sano e disintossicato (ma privo del task assassino). Questo preserva spesso da rovinose perdite di dati o, quantomeno, permette di registrare il lavoro fatto prima di dare libero sfogo al Guru.

In realtà vi sono alcuni particolari casi in cui anche GOMF è impotente: si tratta di condizioni d'errore gravissime e a volte "invisibili" agli occhi di GOMF. Questo non toglie comunque nulla all'utilità del programma.

Per renderlo attivo è sufficiente installarlo nella startup-sequence con...

Run GOMF1.0

Bisogna notare che GOMF è un programma continuamente aggiornato dal suo autore. Le prime versioni erano di pubblico dominio ma poi la HyperTek ha acquistato i diritti su questo programma che è diventato commerciale. La versione di pubblico dominio (e quindi liberamente utilizzabile) più aggiornata dovrebbe essere la 1.0, ma oramai le continue revisioni hanno portato alla versione 3.0. La HiperTek costruisce e distribuisce anche un accessorio hardware: The GOMF Button. Dovrebbe permettere di salvare tutti i dati prima di un qualunque tipo di Guru Meditation. Purtroppo l'accessorio non è ancora distribuito in Italia, ma appena risulterà commercializzato su vasta scala (magari per corrispondenza) provvederemo a parlarne approfonditamente. State comunque tranquilli: con l'aiuto delle nuove release del sistema operativo, e con qualche piccolo strattagemma, il Guru smetterà definitivamente di importunarci.

BASIC E MULTITASKING

Come si può attuare il multitasking da Basic? Vi sono diverse tecniche da utilizzare per simularlo anche all'interno di un programma Basic (Event trapping, interrupt...).

La più semplice ed immediata è senz'altro quella consistente nel... lanciare due interpreti di AmigaBasic!

Se, infatti, dopo aver lanciato AmigaBasic clickiamo nuovamente sulla sua icona, avremo altre due finestre (List e Output) a disposizione per realizzare un altro programma. Naturalmente saremo liberi di lanciarli contemporaneamente senza problemi.

Questo tipo di multitasking è, in realtà, molto rozzo e ce ne possiamo accorgere dal fatto che i programmi di ogni singolo interprete viaggiano ad una velocità decisamente ridotta. Ciò nonostante può ugualmente tornare molto utile avere due editor a disposizione: possiamo confrontare due listati, apporre modifiche ad uno senza perdere il listato originale nell'altro, ma, soprattutto possiamo ritagliare parti di codice da un editor e incollarle nell'altro. Tenetelo presente, può farvi risparmiare molto tempo prezioso.

PROGRAMMI BASIC DA CLI

Se desideriamo lanciare un programma Basic da Cli, possiamo renderci conto che non è sufficiente digitarne il nome. Il Dos, infatti, non riconosce un programma Basic e, non disponendo dell'interprete, emette un messaggio di errore informandoci che il file specificato non è in formato eseguibile.

La corretta procedura consiste, quindi, nel lanciare l'AmigaBasic ed indicare, di seguito, il nome del programma desiderato. Per lanciare da Cli il programma Prova, ad esempio, dobbiamo impartire...

```
AmigaBasic Prova
```

Il sistema provvederà a caricare l'interprete e, subito dopo, il programma "Prova", mandandolo direttamente in esecuzione. Il programma può risiedere anche all'interno di una sottodirectory o addirittura in un altro disco. In questo caso è sufficiente indicare il path (= percorso) completo prima del nome del programma.

Con quest'ultima osservazione ci salutiamo. L'appuntamento rimane fissato, come sempre, tra trenta giorni in edicola.

CARA AMIGA, TI SCRIVO

Da questo numero una nuova rubrica, specifica per il pianeta Amiga

di **Roberto Ferro**

SPRITE IN AMIGABASIC

• Posseggo un Amiga 500 e ho notato che, con l'AmigaBasic, gli sprite ed i bobs non si muovono linearmente come, ad esempio, nel C/128, ma a scatti. Come è possibile fare altrimenti?

(Brachino Luca - Montefiascone)

□ Il problema riscontrato è uno dei difetti più gravi contenuto nell'interprete Basic di Amiga. La gestione di sprite e bob, infatti, viene fatta seguendo una procedura piuttosto grossolana: alla richiesta di aumentare la velocità di movimento dello sprite (OBJECT.VX o OBJECT.VY) l'interprete risponde facendo compiere agli sprite salti di 3, 4 o più pixel in modo da percorrere una stessa distanza in minor tempo. E' chiaro che questo fatto provoca una durezza di movimento praticamente inaccettabile.

In realtà Amiga è capace di muovere uno sprite velocemente e fluidamente (il Blitter è lì per questo), ma alla Microsoft (software house autrice di AmigaBasic) il particolare deve essere sfuggito(!). Scherzi a parte, la supposizione più attendibile è che essendo l'AmigaBasic la versione per Amiga del classico Microsoft Basic per sistemi MS-DOS (che non hanno la benchè minima possibilità di gestire sprite ed affini) le routine per gli sprite sono state scritte ex novo, ma senza cura eccessiva.

Tutto sommato è un vero peccato perchè il set di comandi OBJECT è veramente completo e funzionale. E' facile capire come l'unica soluzione sarebbe quella di manomettere le routine dell'interprete responsabili del movimento degli sprite, ma siccome si tratta di una soluzione assai poco praticabile (anche se teoricamente non impossbile) l'unica alternativa è quella di scrivere routine in C od in Assembly per gestire gli sprite.

In questo caso, infatti, le routine scritte in C (ed organizzate in "librerie") possono esser richiamate da Basic come se fossero sottoprogrammi (con i comandi LIBRARY e CALL), evitando di dover riscrivere il codice in un altro linguaggio.

Tale soluzione non è però di facile realizzazione, e può essere adottata a patto di conoscere in maniera approfondita il System Software di Amiga. Conoscenza, questa, attualmente posseduta solo da pochi esseri umani.

ASSEMBLER PER AMIGA

• Avrei intenzione di programmare l'Amiga in linguaggio macchina e vorrei quindi informazioni sulle istruzioni mnemoniche del 68000 e sulle tecniche necessarie alla sua programmazione.

(Da alcune telefonate)

□ Come è noto, all'interno di Amiga esiste il microprocessore 68000 che governa il funzionamento della macchina, così come il 6510 governa il C/64.

Purtroppo tra i due micro non esistono molte affinità; ne consegue che un discorso sull'assembly dell'Amiga dovrebbe necessariamente cominciare dalle nozioni di base.

Bisognerebbe, inoltre, spiegare le differenze tra un microprocessore ad 8 bit ed uno a 16 / 32 bit, tra uno che viaggia a 1-2 Mhz ed uno che corre a 8 Mhz ed altro ancora (livelli di interrupt, tipi di stack, numero di registri...).

Si tratta, in definitiva, di un discoso complesso che presuppone una certa cultura di base sul pianeta Amiga. Bisogna tener conto che non basta saper programmare il 68000 per programmare Amiga, in quanto è necessario saper sfruttare le risorse dei coprocessori, conoscere la gestione delle periferiche, imparare ad operare in un ambiente multitasking e molto altro. Se, nonostante tutto, siamo ancora convinti di essere all'altezza della situazione, allora

ecco alcuni prodotti che bisogna assolutamente procurarsi:
- Un assemblatore come il Macro Assembler della Metacomco, corredato di indispensabile manuale;
- Un linker (preferibilmente "Blink" della Software Distillery);
- Un manuale sull'assembler MC68000;
- La serie dei Reference Manual di Amiga della Addison Wesley, oppure gli Amiga Programmer's Handbook Volume I e II (purtroppo tutto in americano);
- I sorgenti assembly di alcuni programmi di pubblico dominio;
- Tanta, tanta, tanta pazienza e caparbietà.

Purtroppo nel'immediato futuro non prevediamo (a causa dello scarsissimo numero di richieste) di cominciare un discorso sull'assembly di Amiga. In futuro... chissà.

UN LIETO EVENTO

Signore e Signori siamo lieti di annunciare che da questo mese Amiga avrà la sua posta personale.

Tutti i quesiti relativi a tale computer troveranno risposta in un apposito spazio, richiesto a furor di popolo sia dalle numerose telefonate e lettere di incoraggiamento pervenute, sia in base all'esame dei questionari che iniziano a giungere numerosi (velocità postale permettendo).

Sarà quindi attivato un ulteriore, efficiente servizio con lo scopo non solo di porre in risalto le doti di questo magnifico personal, ma anche di evidenziare le vostre richieste, esaudirle, e per scambiare opinioni e punti di vista.

Potete anche inviare semplici osservazioni, riportare le vostre esperienze circa i problemi (e le soluzioni) che avete via via avuto modo di provare (o che non sapete come rintracciare...).

Sarà, insomma, uno spazio in cui tutti possono parlare di tutto, proporre spunti, elogi e critiche. E' chiaro che la funzione principale rimarrarà sempre quella informativa circa gli eventuali problemi riscontrati usando Amiga.

Vi invitiamo, pertanto, a scrivere su tutto ciò che valga la pena di essere scritto, non dimenticando che ogni lettera sarà sempre esaminata con attenzione ed esaudita mediante risposta diretta, oppure, indirettamente, per mezzo di apposito articolo.

Prima di lasciare la parola ai lettori, vi invitiamo a NON inviare francobolli per risposte personali, nè tantomeno di sprecare denaro in inutili raccomandate: la posta viene esaminata indipendentemente dalla modalità di invio.

SE L'AMIGABASIC VA STRETTO

• Come si può ampliare l'area di lavoro del Basic di Amiga?
(Alessandro Rolandi - Roma)

□ Per aumentare la quantità di memoria destinata ai programmi, esiste il comando CLEAR descritto nel capitolo Memory Management. In breve la sua sintassi è...

CLEAR, spazio per i dati, stack

...dove lo spazio per i dati è quello che contiene il programma, le sue variabili ed un buffer per l'accesso ai file. Per default questa quantità è fissata a 25000 byte.

Lo stack è, invece, la zona riservata a mantenere il controllo di tutte le istruzioni iterative come FOR...NEXT, WHILE / WEND oppure per conservare l'indirizzo di chiamate delle GOSUB. L'ampiezza dello stack non può essere inferiore a 1024 byte. La memoria restante viene normalmente definita "heap memory" e sarebbe il "cumulo" di memoria libera dove il sistema alloca i piani di bit per gli screen e le window, oppure dove vengono elaborati i dati per le istruzioni SOUND e WAVE. Quest'ultima zona di memoria, al contrario delle altre, non è di esclusiva proprietà di AmigaBasic, ma può essere condivisa (e quindi consumata) da tutti gli altri task correntemente attivi.

Un altro effetto sortito dall'istruzione CLEAR è quella di azzerare tutte le variabili in memoria, e quindi, a meno di non farne un uso particolare, è sempre conveniente collocarla in testa al programma, prima di ogni altra assegnazione, onde evitare inspiegabili bug.

AMIGA AL BAR

• Sono in programma conversioni per Amiga di giochi da bar?
(Ignoto)

□ A parte il fatto che gran parte della produzione videogamica per Amiga (e per qualunque altro home) è frutto di una più o meno chiara ispirazione ai giochi da bar (tra i molti, un esempio che sovviene è certamente "Rolling Thunder"), la domanda che è opportuno porsi è: "Sono in programma conversioni per le Sale Giochi dei game Amiga ?".

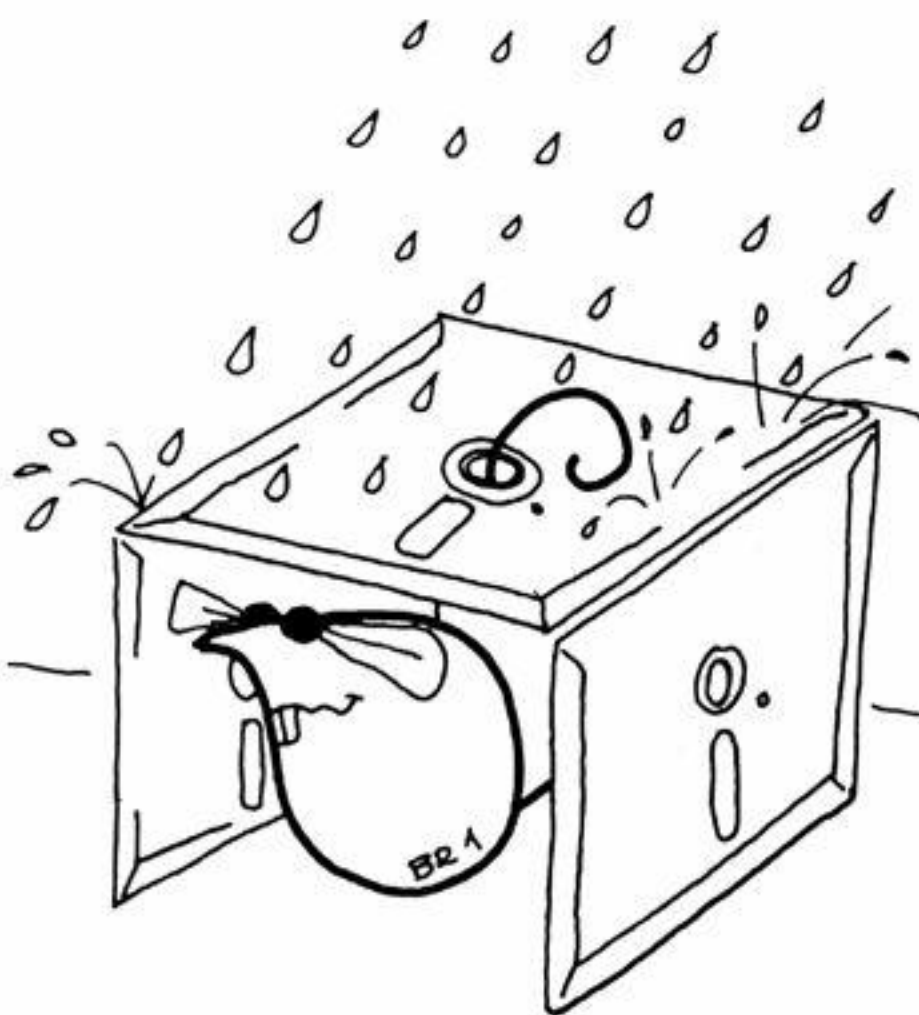
Questo perchè da un po' di tempo pare che alcune importanti produttrici americane di giochi da bar adottino, come piastra madre dei videogiochi, la piastra dell'Amiga che, con i suoi coprocessori, sembra sopravanzare di gran lunga le attuali circuiterie per videogiochi.

E' facile prevedere che se il fenomeno dovesse diffondersi (ma forse i produttori di giochi da bar si affideranno a qualcosa di ancora superiore che adotti la tecnologia CDI - Compact Disk Interattivo) tra non molto potremmo vedere le stesse realizzazioni dei bar sul video di casa nostra, a tutto vantaggio delle nostre tasche.

STRANEZZE NEL WORKBENCH

• Ho notato che impartendo, da CLI, il comando "LoadWB - debug", invece del semplice LoadWB compare, oltre ai soliti, un menu privo di titolo che mostra due scelte: debug e flushlibs. Qual'è la loro funzione?
(Andrea Azzoni - Milano)

□ Se ci troviamo in ambiente CLI, ma non disponiamo del Workbench, significa che siamo entrati in CLI, al momento del boot, mediante un CTRL-D; possiamo tuttavia usare il comando LoadWB per richiamarlo.

La documentazione ufficiale su questo comando (invero molto scarsa) dice che esso non ha parametri. In realtà, come ci fa notare il nostro lettore, un parametro c'è: -debug.

Impartendo LoadWB con tale parametro (non importa se è già presente il Workbench o meno) otterremo il classico screen con i tre menu; spostandoci, però, ancora a destra con il cursore, sarà possibile osservare un menu senza titolo con le due scelte, debug e flushlibs.

Si tratta, probabilmente, di un menu creato mentre il Workbench era ancora in fase di realizzazione, per sperimentare e richiamare particolari funzioni. E' probabile, pertanto, che i progettisti, al termine del loro lavoro, abbiano provveduto a disattivare le due funzioni dal Workbench, ma si sono evidentemente dimenticati di riscrivere anche il comando che lo richiamava, lasciando la possibilità di visualizzare i due menu. Questi, anche se selezionati, non sortiscono nessun effetto, ma a volte inchiodano il sistema.

Quella segnalata è forse un'altra delle stranezze di Amiga, tra cui ricordiamo l'impronta della zampa di cane sull'involucro del modello A-1000, i nomi richiamabili sulla barra del Workbench (con Alt + Alt + Shift + Shift + F1 / F10, e i messaggi romantici delle Preferences). Chissà che cosa ci riserva il futuro.

RIMANDATO IN LINGUA STRANIERA

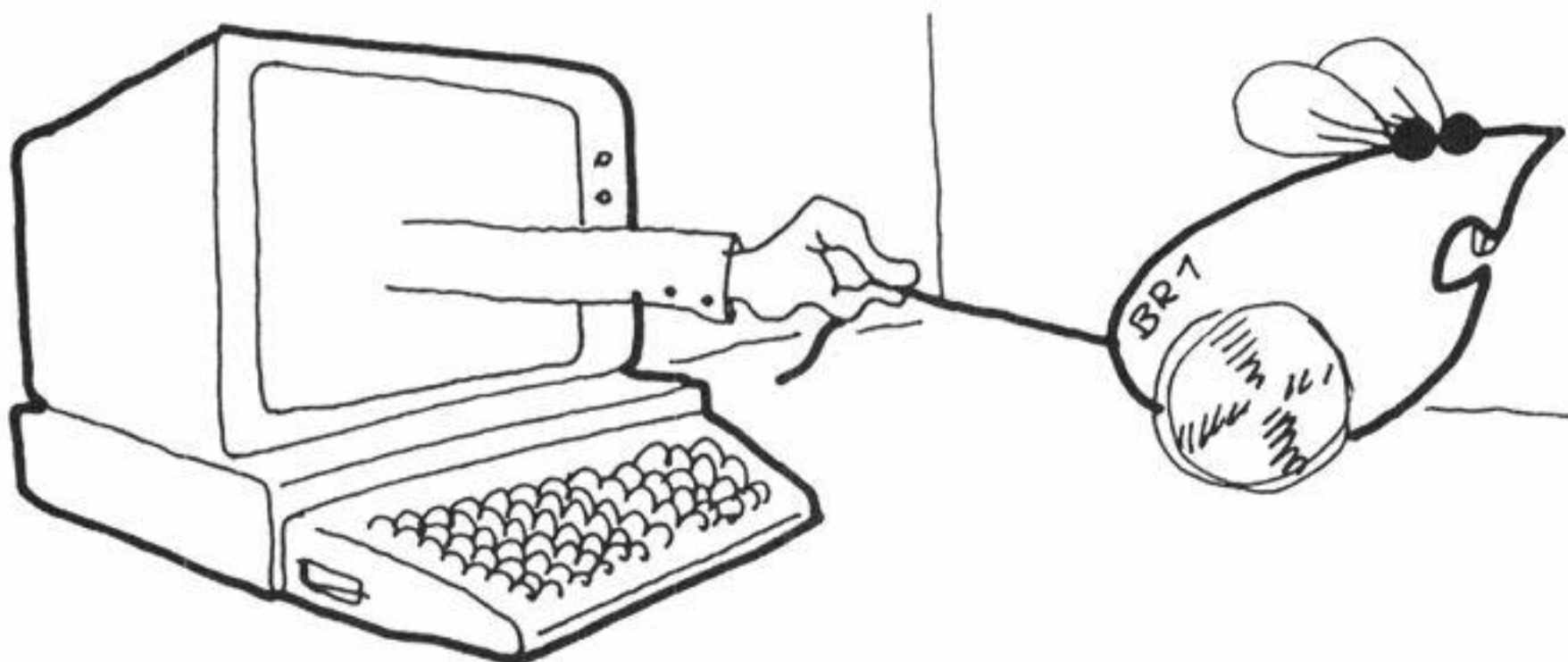
• Poichè sono molto interessato al programma Cli-Voc per Amiga apparso sul numero 52 di C.C.C (e non sono riuscito a farlo girare) vi prego di individuare l'errore nel listato che vi invio su disco e carta.
(Marco De Poi - Trezzo s/Adda)

☐ Sul numero di aprile è apparso un file batch che permetteva la gestione molto rapida e funzionale di un rudimentale vocabolario. Il programma era composto da due file batch, Starter e Cli-Voc.

Nel listato inviato vi sono ben tre errori. Il più grave è l'omissione dei due punti (:) dopo cd Df0.

Come è noto, infatti, assegnando come parametro, ad un comando AmigaDOS, un nome qualsiasi, questo viene considerato come il nome di un disco, oppure di un file.

Se, invece, lo facciamo seguire dal carattere di doppio punto (:), viene considerato come uno dei device collegabili all'Amiga (DF0:, DF1:, DF2:..., PRT:, PAR:, RAM: e così via). Un altro errore è l'aver scritto verso la fine del file "joint" invece del comando "join".

L'ultimo errore è forse il più perdonbile, in quanto molto sottile: all'inizio del file Cli-Voc, dopo la direttiva .key, bisognava scrivere i parametri separati da una virgola ma SENZA spazi, come invece risulta sul disco inviato.

Nell'articolo citato, in effetti, tale particolare non era specificato; provvediamo ora, sottolineando che il listato pubblicato è assolutamente privo di errori ed eventuali malfunzionamenti traggono origine da una errata digitazione.

Ne approfittiamo anche per ricordare che i possessori di un solo disk drive devono tassativamente seguire le procedure descritte nell'articolo stesso, mentre chi ne ha due può utilizzare il batch nel modo seguente:
- eseguire il boot di Amiga da Workbench;
- spostare il Workbench in DF1: e inserire il disco con Cli-Voc in DF0:
- impartire, da CLI, "EXECUTE DF0: starter"
- ignorare la richiesta di inserimento disco e procedere come di consueto.

PERIFERICHE E COMPATIBILITA'

• Che differenza c'è tra il drive A-1010 e l'A-2010? L'Amiga 500 è compatibile PC AT?
(Alberto Serra - Cagliari)

☐ I due drive citati sono entrambi da 3.5 pollici, hanno la stessa meccanica e le stesse capacità. La differenza sostanziale consiste nel fatto che il modello A-1010 viene venduto con un suo cabinet in modo da essere utilizzato come drive esterno collegabile sia ai modelli Amiga, sia ai PC. Il suo prezzo è quindi maggiore del modello A-2010 che, invece, è "nudo" ed è predisposto per essere inserito nell'apposito spazio per il secondo drive sull'Amiga 2000.

La compatibilità con lo standard XT / AT (sistema Ms-Dos) è possibile, almeno in via teorica. E' da notare, tuttavia, che attualmente la Commodore non ha proposto schede specifiche per l'emulazione XT/AT per Amiga 500 (e forse non ne verrano proposte neanche in futuro; non avrebbe senso fare concorrenza al modello 2000, appositamente predisposto). L'unica alternativa è quindi la blanda (ma a volte preziosa) emulazione software via Transformer 1.2.

Ci sarebbe anche una terza via per l'agognata(?) compatibilità: l'acquisto di un apposito cabinet da sistemare sotto l'Amiga 500, collegato ad esso, in grado di ospitare le schede del 2000 (A-2088 / A-2286 ed altre). Di recente abbiamo visto allo SMAU un cabinet del genere (presente da tempo negli Stati Uniti), ma non abbiamo ancora avuto modo di dare uno sguardo più da vicino allo scopo di esprimere un giudizio sulle reali possibilità di utilizzo e convenienza economica.

Prima di concludere vorremmo invitarti ad una riflessione: pensi realmente che, disponendo di un Amiga, si senta la mancanza di un sistema MS-DOS? In caso affermativo, non sarebbe più utile vagliare la possibilità di acquisto di un PC compatibile intero e non su scheda? Se la Commodore non abbassa i prezzi delle sue schede (a cui bisogna aggiungere il costo, seppur contenuto, del cabinet) non vedo la convenienza di "imPIASTRIcciare" il tuo Amiga 500.

DRIVE DA 5 e 1/4

• Esiste la possibilità di installare un drive esterno da 5 pollici e 1/4 per Amiga ?
(Bozzoli Massimo - Bologna)

☐ La gestione dei drive, da parte di Amiga, avviene attraverso routine software contenute nello stesso sistema operativo. Questo insieme di routine viene chiamato trackdisk.device e governa, appunto, le modalità di accesso al disco.

Questo significa che, all'interno del disk drive, non c'è praticamente alcun software (come, invece, accade per i modelli colle-

gabili al C/64) che decide, ad esempio, di formattare un disco ad 1 Megabyte piuttosto che a 500 Kilobyte.

Per il disk drive risulta quindi indifferente effettuare un certo numero di giri piuttosto che un altro, spostare la testina di una quantità minima o massima; è il trackdisk che decide tutto, tant'è vero che esistono in commercio programmi in grado di leggere dischi da 3.5 in formato MS/DOS, Macintosh e Atari utilizzando drive Amiga.

In via teorica, pertanto, è possibilissimo collegare un drive da 5.25 ad Amiga; è sufficiente "solo" che il trackdisk lo riconosca. Purtroppo, almeno attualmente, ciò non accade e pertanto l'operazione risulta impossibile. Per realizzare un simile adattamento è praticabile l'adozione di un software di emulazione, come il Transformer o l'Emulatore C/64. In questo caso, infatti, anche se con metodi differenti, non è più Amiga a governare i disk drive, ma lo stesso software di emulazione, il quale si occupa sia di inviare al drive segnali "comprensibili", sia di trasformare i segnali in arrivo dal drive in modo che Amiga li riconosca. Si tratta, insomma, di una vera e propria interfaccia software che si sostituisce al trackdisk.device.

VIRUS E ANTIVIRUS

• Siamo dolenti di informare dell'esistenza di un nuovo virus: Byte Bandit in 9.87. Si manifesta con un blocco totale della macchina a cui si può rimediare solo con lo spegnimento.
(Club Commodore Computer - Finale Ligure)

Quanto durerà la storia dei Virus? Probabilmente ancora a lungo. Appena viene fuori l'antivirus di un "ceppo", ecco che ne sbuca uno nuovo, di solito più subdolo del precedente.

Le conclusioni che possiamo trarre per il momento non sono allegre, ma vale ugualmente la pena di parlarne.

Innanzitutto il virus citato non differisce dagli altri per ciò che riguarda la posizione sul disco. Si alloca nei boot block ed è quindi facilmente rintracciabile con un disk editor. Entra nel sistema solo se è presente sul disco con cui abbiamo avviato la macchina, ma non ne "esce" con il reset.

Ciò significa che se accendiamo Amiga, e lo avviamo con un disco sano, il sistema rimarrà sano fino al primo reset. Se dopo di esso uilizziamo un disco infetto non c'è reset che tenga: l'Amiga è infettato.

Unico rimedio è lo spegnimento della macchina. Per quanto riguarda la propagazione, ci troviamo di fronte ad uno dei più pericolosi strattagemmi che abbiamo personalmente avuto modo di osservare: il Byte Bandit si insinua nel trackdisk.device ed interviene ogni volta che inseriamo un disco per l'identificazione. Esso lo esamina e, se lo trova sprotetto, vi si riproduce. Tale particolarità presenta un'altra allarmante conseguenza: rende assolutamente inutile adoperare il comando INSTALL per riformattare il boot block perchè, non appena è finita l'azione di INSTALL, il virus provvede subito a riprodursi, vanificando il tentativo di "depurazione".

Quando il virus entra in funzione oscurando il video e bloccando la macchina, possiamo tuttavia riprendere il nostro lavoro attraverso un piccolo trucco. Questo consiste nel premere, consecutivamente, i tasti Alt + Amiga + Spazio + Amiga + Alt tenendoli premuti. Potremo, in questo modo, riprendere il lavoro interrotto, avendo tutto il tempo per salvare l'eventuale attività in corso e spegnere il computer.

Attenzione: uno dei dischi che sicuramente conterrà il virus sarà proprio quello in cui avete effettuato il salvataggio. Riaccendendo, però, Amiga con un disco boot sicuramente sano, potrete copiare i file che interessano e INSTALLare i dischi infetti. Naturalmente, per riconoscere se un disco è infetto o meno, dovete usare un disk editor che, con il dump ASCII del blocco zero, vi informerà sulla presenza del virus. Potete, ovviamente, anche usare il comando...

INSTALL df0: CHECK

...che è presente tra i nuovi comandi del Workbench v1.3. Ciò dimostra come la stessa Commodore si sia mossa in aiuto degli utenti.

DIVENTA IL DOTTORE DEL TUO AMIGA

Se un virus ha infettato i tuoi dischetti, ecco un sistema per curarli; ed un modo di applicare il linguaggio "C"

di **Luigi Callegari**

Ormai si parla dappertutto di virus per Amiga, persino al TG1 e sulle riviste estere. In breve ricordiamo che un virus è un programmino, quasi sempre in Assembly, in grado di riprodursi da un dischetto all'altro, senza che l'utente se ne accorga, di rovinare eventuali dischetti con tracce standard in modo da non consentire operazioni di scrittura.

Alcuni virus, inoltre, bloccano dopo un certo lasso di tempo il computer, presentando uno schermo irrimediabilmente verde.

Solitamente, tutte le trattazioni ed i programmi in circolazione, prevedono solo il virus della svizzera SCA, ma esistono almento altri due virus per Amiga, che sfuggono a tali programmi specifici.

Abbiamo dunque deciso di scrivere di nostro pugno un programma in "C" (presente anche nel nostro periodico su disco, "Amigazzetta") che consente di visualizzare qualunque tipo di Virus eventualmente presente su di un dischetto.

Le premesse poste all'origine sono semplicemente due: un virus, se presente su disco, "deve" essere sempre nel cosiddetto "bootblock", ovvero nei primi due blocchi di dati di 512 byte ciascuno. Ciò gli permette di riprodursi in memoria al momento dell'inserimento del dischetto (al posto di Workbench) e di prendere il controllo del sistema.

Inoltre un virus ha sempre un messaggio ASCII che presenta sul video periodicamente, di solito dopo un certo numero di reset (16 per lo SCA), oppure semplicemente di copyright del virus stesso (Byte Bandit).

Il nostro programmino in C sfrutta le funzioni di AmigaDOS per leggere i contenuti del bootblock, poi visualizza su schermo (o stampa su carta) tutti i caratterii ASCII trovati. Nel caso sia presente un virus, si vedranno messaggi sospetti nell'output (vedi esempi).

COME USARE LA RICETTA

Il programma è stato scritto con Lattice C (versione 4.0). Dovrebbe essere facilmente trasportabile anche sul Manx Aztec C e, sicuramente, funziona anche sulle più vecchie versioni del compilatore (3.10). Il linking può essere fatto con ALINK (Metacomco / Commodore) o con BLINK (software distillery).

Per scrivere il programma si può usare l'editor di sistema ED oppure una qualunque utility di videoscrittura con uscita ASCII standard (MicroEmacs, TxED). Supponendo che il file sorgente sia stato chiamato VIRUSCAN.C compileremo con...

LC -vbr Viruscan

```
ESEMPIO DI OUTPUT DEL PROGRAMMA VIRUSCAN DI UN DISCHETTO INFETTATO
DAL VIRUS DELLA SCA:



 VIRUSCAN: Scansione del drive DF1:

DOS7CHW!AC0<"QNC,yN @ hpNu,y9fB.a<K;¦`p2aF`aA:g#:-H:Nu-¦>.A"B@rXQF@0Nu$f(gNBNDOSf0-y:
BHKA g(IaL?Nuyy09@@faR"L3¦,yN8"L3¦#¦$#¦(#¦,,yN8"L3¦,yN8Nu"LB$3¦,yN8GCB,yN#"K,yN:A'Hp2
<@4<,yNz+¦;¦pdaEA#0<BQ;¦.u;¦;¦8;¦B;¦B;¦"KBrQ,yN"KBgP JE,yNt2<paJ;AA"QBa6tpa.A";AQBa`R
 y+h&;¦Nu@-f-gQNupbtuQ*graphics.librarydos.library Something wonderful has happened-Y
our AMIGA is alive !!!Aand, even better...PPSome of your disks are infectedn2Zby a VI
RUS !!!x2Another masterpiece of22The Mega-Mighty SCA !!nNA!SCA!SCA!SCA!SCA!SCA!SCA!SC
A!

 VIRUSCAN: Lavoro eseguito!
```

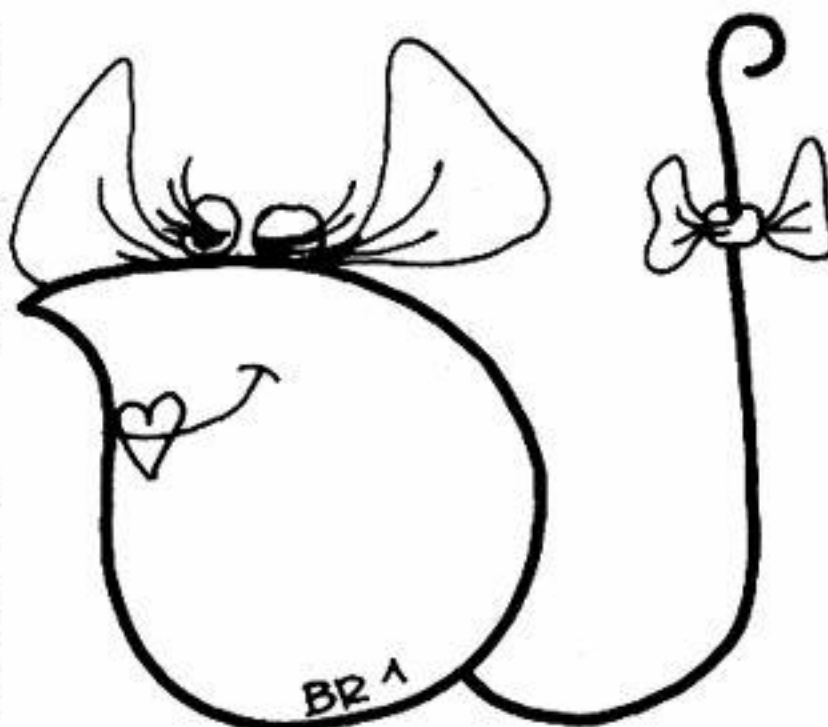
...poi eseguiremo il linking con:

BLINK FROM lib: c.o + viruscan.o TO viruscan LIB lib: lc.lib + lib: amiga.lib CHIP ND SC SD

Per usare il programma ottenuto, che si chiamerà "viruscan", basta copiarlo sul nostro disco di lavoro Workbench (che speriamo non sia infettato!); per usarlo da CLI si digiterà:

VIRUSCAN n

...dove "n" è un numero compreso tra zero e tre, indicante il numero del drive che contiene il dischetto da verificare.Il programma produrrà sul video la stampa di tutti i caratteri ASCII presenti nell bootblock. Se vi è un virus si vedranno scritte intelligenti(!).

Per inviare l'uscita su stampante, si usa il solito sistema di redirezione; la battitura di...

viruscan >PRT: 1

...,ad esempio, scrive l'output del bootblock del dischetto (presente nel primo drive esterno) sulla stampante selezionata da Preferences. Chi possiede un solo drive deve usare viruscan copiandolo nel ramdisk...

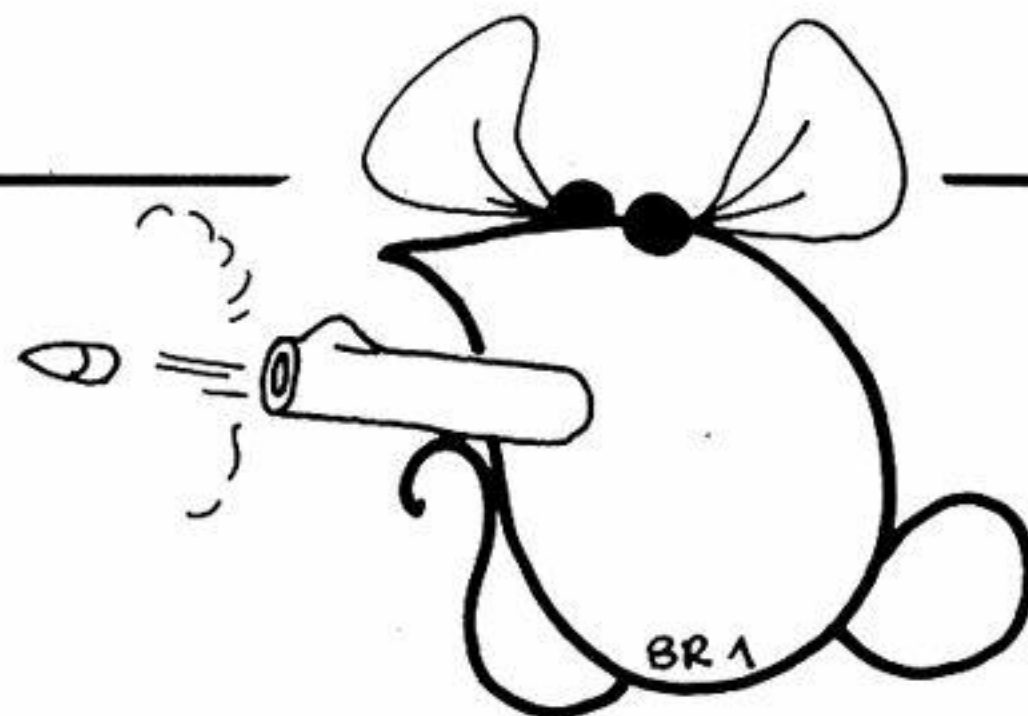
COPY viruscan TO ram:

...poi si inserisce nel drive interno il disco sospetto e si digita:

RAM:viruscan 0

Il programma, come già detto, sarà inserito nel prossimo dischetto AMIGAZZETTA prodotto dalla Systems Editoriale.

```
ESEMPIO DI OUTPUT DEL PROGRAMMA VIRUSCAN DI UN DISCHETTO INFETTATO
DAL VIRUS DELLA BYTE BANDIT:



 VIRUSCAN: Scansione del drive DF1:

DOSA^p`>Virus by Byte  Bandit in  9.87.Number of        copys :,x3@.&0.¦f@AB P"h*IA^E
HfAp A"NA,B  a>aCNJg$ @ h.&13 <LNu <`3@.&H <"<N:A A^CNJg @IK*A( .A A-H.&13NuAC B-H&2J
#IE#J¦¦!¦¦G#K#¦G#KN-@*Nu ),f)<g2<f/ i(A _9ggH@,x3@.&¦&i($K$g &z#K(#¦$ACf$KACxf K2<BdR
QF"+d'AADPgP0¦e0A.&13 zNKUf&KUfKRea*zN3@.&H3KBK9g99@F2<3Q9fg`3L.&13NuAF0¦dP,xaJNu0'df
@getrackdisk.devicedos.library

 VIRUSCAN: Lavoro eseguito!
```

```
/* VIRUS SCANNER - V1.2 di Luigi R. Callegari per Commodore Computer Club
   Visualizza caratteri ASCII in Bootblock, permettendo visione di virus
   File Viruscan.c - Compilato con Lattice AmigaDOS C V4.00 - Blink V7.2
*/

#include <exec/types.h>
#include <exec/io.h>
#include <devices/trackdisk.h>
#include <stdlib.h>
#include <stdio.h>

#define TD_READ   CMD_READ
#define BLOCKSIZE TD_SECTOR
#define NUMHEADS  2

struct Port      *diskport, *CreatePort();
struct IOStdReq  *diskreq,  *CreateStdIO();

BYTE  *diskdata, diskbuffer[BLOCKSIZE]; /* Buffer per il disco */
SHORT error, testval;

void cleanup(s,c)        /* Gestione di errori ed uscita */
char *s; short c;
{
        fprintf(stderr,"\n%s\n",s);
        exit(c);
}

void ReadCylSec(cyl, sec, hd)   /* Lettura random del disco */
SHORT cyl, sec, hd;
{
LONG offset;
diskreq->io_Length    = BLOCKSIZE;
diskreq->io_Data      = (APTR)diskbuffer;
diskreq->io_Command   = TD_READ;
offset = TD_SECTOR * (sec + NUMSECS * hd + NUMSECS * NUMHEADS * cyl);
diskreq->io_Offset     = offset;
DoIO(diskreq);
if ( diskreq->io_Error !=0 )
        cleanup("Errore dalla periferica!",2);
return;
}

void MotorOn()            /* Accende il motore del drive */
{
diskreq->io_Length  = 1;
diskreq->io_Command = TD_MOTOR;
DoIO(diskreq);
return;
}
```

```
void MotorOff()            /* Spegne il motore del drive */
{
diskreq->io_Length = 0;
diskreq->io_Command = TD_MOTOR;
DoIO(diskreq);
return;
}

void  main(argc,argv)   /* Funzione principale di gestione */
int argc;
char **argv;
{
register int i, j, c, ndisk, error;

if (argc != 2)  cleanup("Uso: Viruscan n (n= 0,1,2,3)",3);

ndisk = atoi(argv[1]);
if ( ndisk < 0 || ndisk > 3 )
        cleanup("VIRUSCAN: Numero drive non compreso tra 0 e 3!",4);
diskdata  =  diskbuffer;
diskport  =  CreatePort(0,0);
diskreq   =  CreateStdIO(diskport);
if ( (error = OpenDevice(TD_NAME,ndisk,diskreq,0) ) != 0 ) {
        cleanup("VIRUSCAN: Non riesco ad aprire la periferica!",error);
}

printf("\n\033[1m VIRUSCAN\033[0m: Scansione del drive DF%d:\n\n", ndisk);
MotorOn();
for ( i = 0; i < 2 ; i++) {
        ReadCylSec(0, i, 0);
        for ( j = 0 ; j < BLOCKSIZE ; j++) {
                if ( ( c = (int)diskbuffer[j] ) > 31 && c < 128 )
                        putchar(c);
                }
        }
MotorOff();                CloseDevice(diskreq);
DeleteStdIO(diskreq);      DeletePort(diskport);
printf("\n\n%s\n\n\n","\033[1m VIRUSCAN\033[0m: Lavoro eseguito!");
}
/*         -------------------- End Of Program --------------------  */
```

CORSI E RICORSI

La più importante caratteristica dei linguaggi evoluti, la "ricorsione", realizzata in Turbo Pascal; ma anche in Basic

di **Claudio Baiocchi**

Tra le possibilità più interessanti offerte dal linguaggio Pascal, un posto di rilievo spetta senza dubbio alla facilità con cui si possono trattare problemi di natura ricorsiva. In questo articolo illustreremo, con qualche esempio, il concetto stesso di ricorsività; vedremo in particolare che scrivere programmi ricorsivi è estremamente facile in ambito Pascal, mentre in ambiente Basic richiede un po' di acrobazie; discuteremo infine pregi e difetti dell'approccio ricorsivo che, come vedremo, risulta spesso poco efficiente.

RICORSIONE

La tradizione vuole che ogni discorso sulla ricorsività parta dalla (un po' barbosa) nozione di fattoriale; c'è un ottimo motivo per rispettare la tradizione anche se, come vedremo, si tratta di una... falsa partenza. Ma procediamo con ordine, ricordando che i matematici indicano con N! (si legge: N fattoriale) un numero che ha senso solo per N intero e che è definito dalla relazione...

N! = N*(N-1)!

SCHEDA TECNICA

Software didattico per ricerche di tipo matematico

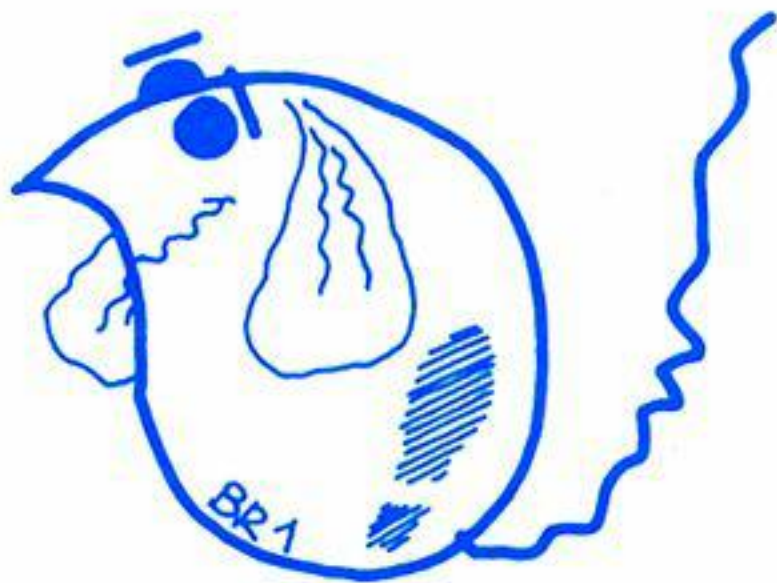
Hardware richiesto: C/128 (per listati specifici); qualsiasi computer Basic (per i listati più semplici); qualsiasi computer capace di supportare il Pascal o il Turbo Pascal

Richiede CP/M, e Turbo-Pascal (C/128)

Richiede il disk drive

Ideale l'uso di un monitor a colori

Consigliato agli esperti

I due programmi Basic più lunghi, pubblicati in queste pagine, sono contenuti nel disco "Directory" di questo mese.

...se N è maggiore di zero, altrimenti...
0! = 1
...se N=0.

In realtà, al livello di definizione, si cerca di rispettare delle esigenze estetico - formali che, dal punto di vista pratico, sono difficili da "leggere": in pratica conviene far riferimento alla formula:

N! = 1 x 2 x 3 x...x N

...da completare poi con i casi speciali 0!=1, 1!=1. Si noti la stretta analogia con la nozione di potenza ad esponente intero positivo, che si definisce (intendendo con "Exp" il significato di "Elevato a") tramite:

A exp 0 = 1
A exp N = A x A exp (N-1)

...ma che poi, in pratica, è vista come una generalizzazione della nozione di quadrato e di cubo:

A exp N = A x A x...x A (N volte)

...estendendo poi tale nozione con i casi speciali A exp 1 = A, A exp 0 = 1.

Un terzo argomento, anch'esso classico in ambito ricorsivo, è costituito dalla "successione di Fibonacci": si tratta di una famiglia di numeri, che indicheremo con Fi(0), Fi(1), Fi(2),... definiti da...

Fi(0)=0; Fi(1)=1; Fi(N)=Fi(N-1)+Fi(N-2)

...per N maggiore di 1.

Non è facile, in questa sede, spiegare a cosa serve tale successione, anche se essa gioca un ruolo importante in varie questioni di Matematica e di Informatica.

In Pascal le tre formule viste (fattoriali, potenze, Fibonacci) sono già "pronte per l'uso", come può dimostrare il listato 1.

Lo sforzo da fare per scrivere tale porzione di programma è pressochè nullo; un analogo programma Basic non è impossibile, ma occorre fare un po' più di attenzione: così, ad esempio, per calcolare N! occorre porre dapprima N=N-1, per calcolare (N-1)!; una volta calcolato tale fattoriale occorre moltiplicarlo per il vecchio valore di N, valore che si ottiene ponendo N=N+1. In forma di subroutine si potrà scrivere il listato 2 (in Basic).

Per l'innalzamento a potenza, se non si è interessati a conservare il valore iniziale di N, basta scrivere il listato Basic n.3 mentre, per quanto concerne i numeri di Fibonacci, la cosa sarebbe un po' più complicata, e non la affronteremo.

Si studino attentamente i listati Basic, provando eventualmente a sviluppare "a mano" ciò che fanno le due subroutine se vengono chiamate con N=0, 1, 2, 3: se si riesce a capire bene il meccanismo della ricorsione su questi esempi semplici, sarà poi facile capirlo anche in casi più complicati. E anzi, la semplicità dei programmi fin qui esaminati è l'unico motivo che giustifica la scelta di questi, come primi esempi di problemi ricorsivi: da ogni altro punto di vista la scelta sarebbe totalmente ingiustificata.

In effetti, tornando alle versioni "naive" di fattoriale e di potenza viste all'inizio, una If...Then ed un ciclo For...Next bastano per scrivere un programma iterativo altrettanto efficiente e comprensibile di quello ricorsivo; sia in Basic, sia in Pascal dove, per le regole che governano i cicli, la IF risulta inutile; si confrontino, infatti, i listati 4 (Basic) e 5 (Pascal).

D'altronde, riflettendo un po' di più, ci si accorge che, per quanto riguarda i fattoriali, anche la soluzione iterativa ora presentata non serve praticamente a nulla: già per N = 34 il numero N! è "troppo grande" rispetto alle capacità sia del Basic che del Pascal (si tratta di un numero con 39 cifre) ed un tentativo di calcolarlo genererebbe un messaggio di overflow; in ogni programma che debba fare spesso uso di fattoriali la "buona" soluzione diventa allora la banalissima soluzione "tabulare": si inizializza una volta per tutte un vettore e poi si fa riferimento ai valori di tale vettore. Precisamente in Basic si farà come suggerisce il listato 6, ed in Pascal analogamente.

Per il problema dell'innalzamento a potenza le considerazioni da svolgere sono un po' diverse, e sono sviluppate in dettaglio nel riquadro specifico; per quanto concerne invece i numeri di Fibonacci, il programma Pascal prima citato è elegantissimo, chiarissimo, e... totalmente inefficiente: in Turbo Pascal, versione per C/128, il calcolo di Fi(30) tramite tale formula richiederebbe un tempo di calcolo superiore alle due ore! Nel riquadro è fornita una soluzione iterativa che richiede tempi di calcolo più che accettabili; tuttavia, se non si hanno grossi problemi di penuria di Ram, anche qui è preferibile una soluzione "tabulare" (che mangia ora un po' più di spazio, poichè l'overflow si presenta solo a partire da Fi(185)) usando il listato n.7.

APPLICAZIONI

Passiamo ora ad un problema in cui la ricorsione risulta veramente utile, discutendo il famoso problema delle "Torri di Hanoi".

Secondo una antica leggenda, in un monastero della città di Hanoi i monaci sono da tempo immemorabile dediti ad un compito apparentemente insensato: su tre piedistalli, marcati rispettivamente come Partenza, Transito, Destinazione (e noi useremo le iniziali P, T, D) sono distribuiti 64 dischi di vario diametro; i dischi vengono spostati, uno alla volta, da un piedistallo all'altro, rispettando sempre la regola di non sovrapporre mai un disco grande ad uno più piccolo. Nella notte dei tempi i dischi erano tutti disposti, in ordine decrescente, sul piedistallo P; compito dei monaci è trasportarli tutti sul piedistallo D (la leggenda dice che il termine dell'operazione segnerà la fine del mondo).

Mettiamoci nei panni di un ipotetico Monaco Istruttore che, per istruire i colleghi, insegna loro a lavorare con torri costituite da un numero inferione di dischi; se arriva da noi un monaco che ha già imparato a lavorare con torri da 5 dischi, e vuole imparare a lavorare con torri da 6, cosa gli suggeriamo? Un'occhiata al seguente "foglio di istruzioni" ci fa capire che la cosa non è poi così difficile:

PROMEMORIA PER 6 DISCHI
Per spostare 6 dischi dalla torre U alla torre V, usando la torre W come transito.
(1) Sposta 5 dischi dalla torre U alla torre W nel modo già noto.
(2) Sposta un disco (è il n.6) dalla torre U alla torre V.
(3) Sposta 5 dischi dalla torre W alla torre V.

Naturalmente, se a chiederci aiuto fosse un monaco che sa già lavorare con 30 dischi (anzichè solo con 5) e vuole imparare a lavorare con 31, basterà ricopiare il pro-

NOVITA' DI GRIDO!

MT 81 E' L'UNICA STAMPANTE PROFESSIONALE A L. 299.000 + IVA

- 80 COLONNE A 10 CPI
- 130 CPS IN ALTA VELOCITÀ
- 24 CPS IN ALTA DEFINIZIONE
- COLLEGABILE A HOME E PERSONAL COMPUTER
- MASSIMA SILENZIOSITÀ

Puoi trovare la MT 81 in tutta Italia presso le reti di vendita di: MILANO *- SILVERSTAR LTD SPA - TEL. 02/4996* ■ ***MILANO*** *- ACS ELETTRONICA SPA - TEL. 02/5398721-5694082* ■ ***MILANO*** *- CLAITRON SPA - TEL. 02/3010091* ■ ***TORINO*** *- ABACUS SRL - TEL. 011/6680164* ■ ***VERONA*** *- TELESTORE 2 SRL - TEL. 045/541051* ■ ***TRIESTE*** *- I.B.C. SRL - TEL. 040/733395* ■ ***REGGIO EMILIA*** *- H.S.D. SRL - TEL. 0522/557600* ■ ***BOLOGNA*** *- NON STOP SPA - TEL. 051/765299* ■ ***RAVENNA*** *- S.H.R. ITALIA SRL - TEL. 0544/463200* ■ ***FIRENZE*** *- DEDO SISTEMI SPA - TEL. 055/4360251-4361901-4361902* ■ ***ROMA*** *- ALTEC SRL - TEL. 06/3605943-3615744-3615745* ■ ***NAPOLI*** *- MASTERS INFORMATICA SRL - TEL. 081/7703024-7703025* ■ ***PALERMO*** *- BELCO SRL - TEL. 091/547566-545827*

MANNESMANN TALLY srl
20094 Corsico (MI)
Via Borsini, 6
Tel. (02) 4502850/55/60/65/70

memoria sostituendo, dovunque, 5 con 30 e 6 con 31. Il nostro promemoria è quindi "universale", nel senso che va bene ad ogni livello?

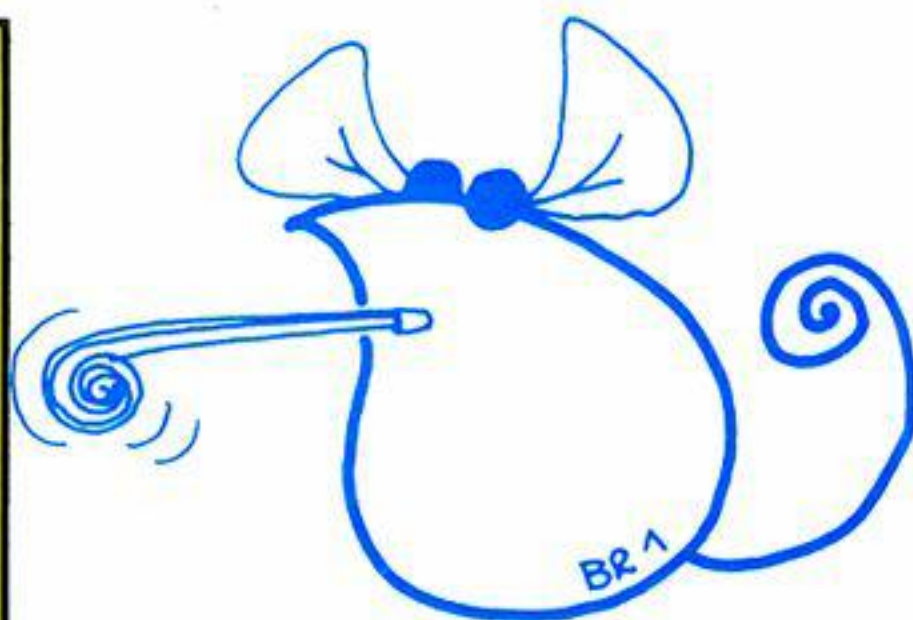
No, non è stato previsto il livello più stupido: se a chiedere aiuto è il novizio, che non sa ancora fare niente, e vuole imparare a lavorare con un solo disco, dobbiamo sopprimere le fasi (1) e (3); ma poi tutto è a posto: un novizio impaziente, che non sa ancora fare nulla e vuole imparare direttamente a lavorare con 6 dischi potrebbe cavarsela usando i Promemoria dal n.6 al n.2, ed il n.1 così modificato.

Nel programma Pascal n.8 la procedura Hanoi è solo una riscrittura del promemoria con un N generico al posto del numero 6.

Si faccia attenzione alla riga iniziale: in generale in Pascal le frasi che iniziano con parentesi aperta - asterisco "(*" e terminano con asterisco - parentesi chiusa "*)" sono dei commenti, analoghi alle Rem del Basic; volendo si può perciò evitare di trascriverli. Tuttavia il Turbo Pascal usa le frasi che iniziano con parentesi - asterisco - dollaro "(*$" come "direttive di compilazione"; sulla versione CPM/80 la direttiva (*$a-*) è essenziale per il buon funzionamento di programmi ricorsivi.

Come tradurre in Basic il programma? Abbiamo già visto che il buon sostituto della procedura è la subroutine; perciò si veda il listato 9.

Studiando "in parallelo" i due listati (e in particolare la procedura Hanoi e la subroutine 150) ci si rende conto facilmente che essi fanno esattamente le stesse cose; solo

```
function fatt(n:integer):real;
  begin if n=0 then fatt:=1
  else fatt:=n*fatt(n-1)
end;
function pot(n:integer;a:real):real;
  begin if n=0 then pot:=1
  else pot:a*pot(n-1,a)
end;
function fi(n:integer):real;
  begin if n<2 then fi:=n
  else fi:=fi(n-1)+fi(n-2)
end;

Listato n.1
-----------------------------------------

990 rem gosub 1000 fornisce f = n!
1000 if n=0 then f=1: return
1010 n = n-1: gosub 1000
1020 n = n+1: f=n*f: return

Listato n.2
-----------------------------------------
1990 rem gosub 2000 fornisce p=a exp n
2000 if n=0 then p=1: return
2010 n = n-1: gosub 2000
2020 p = a*p: return

Listato n.3
-----------------------------------------

990 rem gosub 1000 fornisce f=n!
1000 f=1: if n=0 then return
1010 for x=1 to n: f=x*f: next: return
1990 rem gosub 2000 fornisce p=a exp n
2000 p=1:if n=0 then return
2010 for x=1 to n: p=p*a: next: return

Listato n.4
-----------------------------------------

function fatt(n:integer):real;
var f:real;x:integer;
begin f:=1; for x:=1 to n do f:=x*f;
fatt:=f end;
function pot(n:integer;a:real):real;
var p:real;x:integer;
begin p:=1; for x:=1 to n do p:=p*a;
pot:=p end;

Listato n.5
-----------------------------------------

.....
50 dim f(33):f(0)=1:for x=1 to 33
60 f(x)=x*f(x-1):next
70 rem f(n) fornisce n!
.....

Listato n.6

.....
50 dim fi(184):fi(0)=0:fi(1)=1
60 for x=2 to 184
70 fi(x)=fi(x-1)+fi(x-2):next
80 rem numeri di fibonacci
.....

Listato n.7
-----------------------------------------

(*$a-*)
var nd:integer;
procedure hanoi(n:integer;u,v,w:char);
        (* sposta n dischi da u a v
           usando w come transito *)
  begin
   if n>1 then hanoi(n-1,u,w,v);
   writeln('disco #',n,'da ',u,' a ',v);
   if n>1 then hanoi(n-1,w,v,u)
  end;
begin
write('quanti dischi ? ');
readln(nd);
if nd>0 then hanoi(nd,'P','D','T')
end.

Listato n.8
-----------------------------------------

100 input "quanti dischi";nd
110 if nd<1 then print "pochi !":end
120 if nd>63 then print "troppi !":end
130 u$="P":v$="D":w$="T":n=nd
140 gosub 150:end
150 if n>1 then n=n-1:z$=v$:v$=w$:w$=z$:
     gosub 150:n=n+1:z$=v$:v$=w$:w$=z$
160 print "disco #" n " da" u$ " a" v$
160 if n>1 then n=n-1:z$=u$:u$=w$:w$=z$:
    gosub 150:n=n+1:z$=u$:u$=w$:w$=z$
170 return

Listato n.9
```

che in Pascal è il linguaggio che pensa a quasi tutto, mentre in Basic occorre scrivere esplicitamente le formule di modifica delle variabili, ripristinando poi i valori corretti al "rientro" dalla subroutine dal momento che il Basic non dispone di "variabili locali".

Naturalmente tali listati sono molto "grezzi"; suggeriamo ai lettori volenterosi alcune delle possibili migliorie, tra cui:

- l'output non è entusiasmante: viene solo fornito un arido elenco di mosse, mentre si potrebbe pensare di visualizzare sullo schermo l'evoluzione dei dischi sulle tre torri (su questo punto torneremo tra poco);
- indicando le torri con i numeri 0, 1 e 2, anzichè con le lettere P, T, D, si può pensare di eliminare la variabile W: date U e V, la terza torre si ottiene con la formula 3-U-V; specialmente nella versione Basic ciò accelera sensibilmente le operazioni di "cambio di torre".
- in generale in Pascal ogni chiamata di procedura "mangia" tempo e spazio: occorre costruire da un lato le "variabili locali" della procedura, e d'altro lato dei parametri di collegamento tra la procedura ed il programma principale (ad esempio si deve memorizzare l'"indirizzo di rientro" al termine della procedura).

"DENTRO" LA MACCHINA

In Basic la chiamata di subroutine non mangia spazio per le variabili (non esistono variabili locali) ma l'indirizzo di rientro va comunque memorizzato; e tale memorizzazione, in Basic, è effettuata in una zona molto ristretta della Ram; in termini tecnici si parla di "stack".

Si dia un'occhiata alla riga 120 del listato Basic: il valore massimo 63 imposto al numero di dischi non è dovuto nè a errori di stampa, nè a motivi religiosi o scaramantici (ricordate? risolto il problema con 64 dischi il mondo avrà fine...): si tratta semplicemente del fatto che, nel Basic Commodore, ogni chiamata di subroutine mangia

CALCOLO VELOCE DI POTENZE AD ESPONENTE INTERO

Nell'articolo è illustrato uno schema ricorsivo per il calcolo di potenze A exp N, con esponente N intero.

Si tratta di una nozione molto più elementare di quella di potenza con esponente reale; ma il Basic Commodore non distingue il caso N intero dal caso N reale e fornisce una sgradevolissima segnalazione di errore se si prova a far eseguire (-7) exp 2 o analoga formula; quanto al Pascal, la funzione potenza non esiste(!).

Nell'articolo sono indicati due possibili approcci (uno ricorsivo ed uno iterativo) che presentano, però, lo stesso tipo di inconveniente: il tempo di esecuzione risulta, grosso modo, proporzionale all'esponente N, e cioè: se raddoppiamo il valore di N raddoppia anche il tempo speso dal computer per fornire il risultato. Per ovviare a tale inconveniente bisogna cambiare algoritmo: il metodo che avevamo usato per il calcolo faceva riferimento solo alla definizione, e non sfruttava le proprietà delle potenze che, invece, possono suggerire strade più efficienti; già che ci siamo, tratteremo anche il caso di esponente N negativo (ricordiamo che, per N minore di zero, si pone:
A exp N = (1 / A) exp (-N)

Naturalmente occorre che sia A diverso da zero.

Una delle strade possibili fa uso dei famigerati logaritmi: osservato che...
A exp N = A * Abs(A) exp (N-1)
...e che per il calcolo della potenza di Abs(A) (che è positivo) si può fare uso dei logaritmi, in Basic si può scrivere il listato 13; la versione Pascal è simile.

Un'altra strada, di efficienza temporale analoga, si basa sul fatto che per il calcolo di A exp 4 non sono in realtà necessarie le 3 moltiplicazioni A * A * A * A, ma ne bastano 2: si farà dapprima B = A * A, poi B * B; analogamente per il calcolo di A exp 5 bastano 3 moltiplicazioni, essendo A exp 5 = A * A exp 4; tradurre in programma tale osservazione è tutt'altro che banale, ma ne vale la pena: il programma risultante (listato 14 in Basic e 15 in Pascal) èforse illeggibile, ma efficiente.

```
procedure aggiorna(var ca,cn:real)
var temp:real;
begin temp:=cn; cn:=ca;
ca:=ca+temp end;
```

Listato n.10

```
type fib = 0..184
 (* per n>184 fi(n) e' troppo grande
 ed il suo calcolo genera overflow *)
function fi(n:fib):real; var cn,ca:real; x:integer;
  procedure aggiorna; var temp:real
      begin temp:=cn; cn:=ca; ca:=ca+temp;
      end; (* fine di aggiorna *)
begin ca:=0; cg:=1;
for x:=1 to n do aggiorna;
fi:=ca end;
```

Listato n.11

```
990 rem gosub 1000 fornisce f=fi(n)
1000 if n<0 or n>184 then stop
1010 ca=0:cn=1
1020 if n>0 then for x=1 to n:
     temp=cn:cn=ca:ca=ca+temp:next
1030 f=ca:return
```

Listato n.12

7 byte nello stack; e dopo 63 volte, di spazio non ce ne è più. Si pensi che C/128, C/16 e Plus/4 "reggono" meno dischi; Vic/20 e C/64 ancora meno...; invece l'M-basic in nostro possesso, per C/128 "sotto" CP/M, regge fino a 85 dischi.

Dal punto di vista del problema studiato la restrizione non è grave: già per valori di ND intorno alla quarantina non riusciremmo a campare abbastanza per vedere la fine dell'elaborazione; però si tratta di un punto a favore del Pascal: grazie ad una diversa gestione dello stack, il programma Pascal potrebbe girare con valori di ND largamente superiori al migliaio (ci riferiamo al Turbo Pascal Borland, versione specifica per C/128).

Sorprendentemente il Basic si prende una rivincita nel confronto tra i tempi di esecuzione: su C/128, modo 128 e schermo a 80 colonne, il programma Basic, in modo fast, impiega 75 secondi per fornire la soluzione relativa a ND = 10; il programma Pascal, sempre su C/128 con schermo a 80 colonne, ne impiega 105. Come mai un linguaggio interpretato fornisce prestazioni migliori di uno compilato? Personalmente propendo per la seguente spiegazione: come vedremo tra poco, per ND=10 occorrono 1023 mosse per risolvere il problema; per scrivere l'elenco corrispondente, entrambi i programmi spendono la maggior parte del tempo a far "scrollare" il video; e sul C/128 lo schermo a 80 colonne è gestito meglio dall'8502 (processore usato dal Basic) che dallo Z/80 (processore usato dal CP/M, e quindi dal Pascal). Sullo schermo a 40 colonne le cose peggiorerebbero, perchè il Pascal gestisce comunque uno schermo da 80, visualizzandone solo 40.

EFFICIENZA

Ma torniamo al problema in generale, e chiediamoci: la strada seguita è la migliore possibile? E cioè: non ci saranno metodi più efficienti, che risolvono il problema in un numero inferiore di mosse? Per rendersi conto che il programma lavora "al meglio" osserviamo che per risolvere il problema con 6 dischi, la fase (2) del Promemoria va comunque eseguita; e le regole del gioco richiedono che, per effettuarla, tutti i dischi più piccoli siano stati trasferiti sulla torre W; in altre parole, prima di poter eseguire la fase (2) dobbiamo, in un modo o nell'altro, aver eseguito la fase (1); poi, eseguita la (2), dobbiamo per forza eseguire la fase (3). Ne segue che, se le fasi (1) e (3) sono "eseguite bene", anche il Promemoria risolve il problema nel numero minimo di mosse; e se servono K mosse per risolvere il problema con 5 dischi, ne serviranno K +

RICORSIONE E CONIGLI

Intorno al 1200 un tale Leonardo da Pisa, detto Fibonaccio (cioè: figlio di Bonaccio), si propose di ottenere una formula per valutare il reddito prodotto da un allevamento di conigli. La matematica dei tempi non era molto sviluppata, cosicchè egli studiò il problema in ipotesi semplificative molto rozze; precisamente egli suppose che:

- una coppia di conigli adulti genera ogni mese una coppia di conigli;
- i conigli neonati impiegano un mese per diventare adulti.

Si osservi che, oltre a trascurare effetti di malattie, invecchiamento e morte, sono stati anche soppressi gli aspetti legati al sesso: "coppia" significa "un maschio e una femmina". A questo punto i conti diventano abbastanza semplici: se in un dato mese si hanno CN coppie di neonati e CA coppie di adulti, il mese dopo si avranno CA coppie di neonati e CA+CN coppie di adulti. In Pascal potremmo effettuare il calcolo tramite la procedura "aggiorna" del listato 10.

Se iniziamo con un'unica coppia di neonati (cioè: CA=0 e CN=1) il calcolo dei valori di CA e CN dopo N mesi si ottiene banalmente reiterando N volte la procedura "aggiorna"; i valori assunti da CA al "mese n.0" (di partenza), al mese n.1, n.2 e così via si dicono "numeri di Fibonacci" e coincidono con i numeri che nell'articolo abbiamo indicato con Fi(0), Fi(1), Fi(2),...

I listati 11 (in Pascal) e 12 (in Basic) permettono di valutare Fib(n) seguendo tale idea; un altro modo di calcolare tali numeri è fornito da una loro "rappresentazione esplicita" che però (nonostante si tratti di numeri interi) fa intervenire nel calcolo la radice quadrata di 5; indicando con R tale radice si ha:
Fi(N) = [((1 + R) / 2) exp N - ((1 - R) / 2) exp N] / R

```
990 rem gosub 1000 fornisce p=a↑n
995 rem se a=0, n<0 si ha
    segnalazione di errore
1000 if a=0 then if n>=0 then p=1:return
1010 p=a*exp(log(abs(a))*(n-1)):return
```

Listato n.13

```
990 rem gosub 1000 fornisce p=a↑n
1000 b=a:c=1
1010 m=abs(n):if n<0 then b=1/a
1020 rem deve segnalare
     errore se a=0, n<0
1030 rem la linea 1050 simula
     un comando while
1040 rem evitando i
     lenti "goto all'indietro"
1050 for x=1 to 9e9
1060 q=int(m/2)
1070 if q+q<m then c=c*b
1080 m=q:if m>0 then b:=b*b:next
1090 p=c:return
```

Listato n.14

```
function pot(n:integer;a:real):real;
(* pot(n,a) fornisce il valore di a↑n *)
  function potpos(n:integer):real;
     var z:real;
     begin z:=1;
     while n>0 do
       begin if odd(n) then z:=z*a;
       n:=n div 2; if n>0 then a:=sqr(a)
       end; potpos:=z
  end; (* di potpos *)
begin if n<0 then begin n:=-n;a:=1/a end;
   (* occorre segnalare errore se n<0 e a=0 *)
   pot:=potpos(n)
end;
```

Listato n.15

1 + K per risolvere il problema con 6. Il discorso resta valido qualunque sia il numero di dischi ND, e porta alla formula generale:

nmin = 2 exp ND - 1

in cui, ovviamente, per nmin si intende il numero minimo di mosse.

Tante sono effettivamente le mosse che il programma, sia in versione Basic che in versione Pascal, impiega per risolvere il problema (naturalmente per dare una giustificazione completa si dovrebbe fare uso del "principio di induzione"...).

Visto ciò, possiamo discutere l'efficienza temporale dei nostri programmi: poichè passando da un valore ND al valore ND+1 l'elenco delle mosse da stampare, grosso modo, raddoppia, i programmi sono da ritenere "ben fatti" se, grosso modo, raddoppia anche il tempo di esecuzione (se il tempo triplicasse, o peggio, dovremmo concludere che il programma è fatto male!). Ed in realtà i programmi si comportano esattamente nel modo auspicato: con buona approssimazione, per valori di ND non troppo bassi, il tempo di esecuzione risulta del tipo:

T = C * 2 exp ND

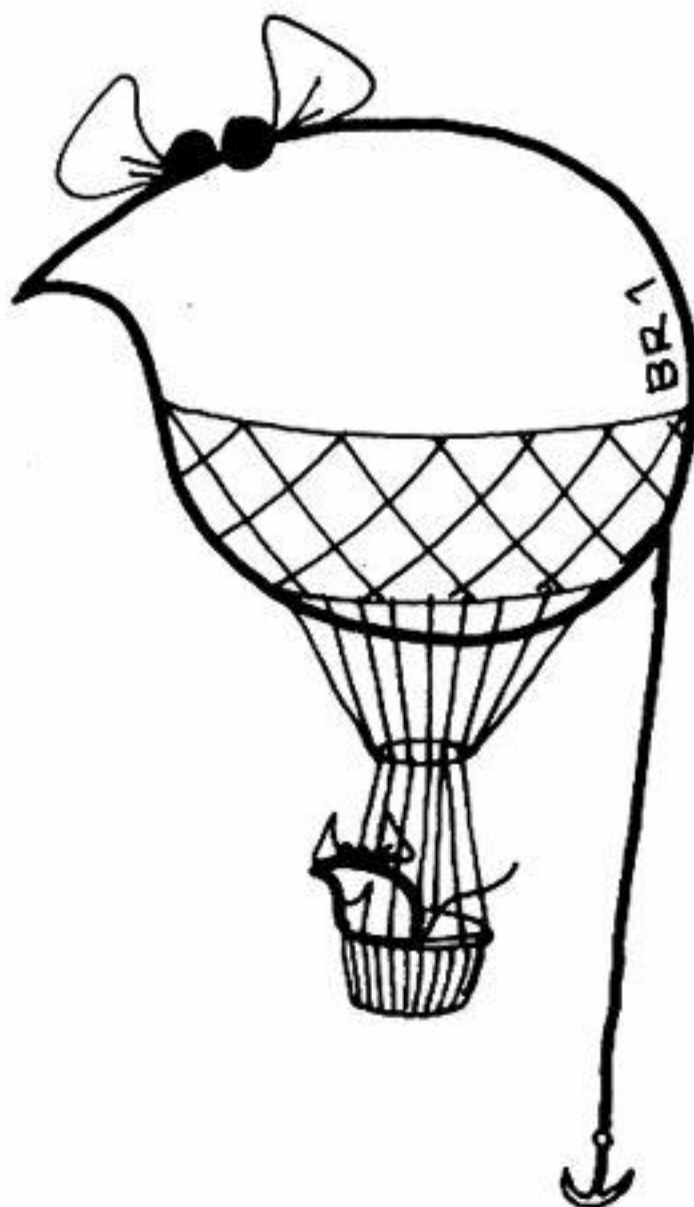
in cui il valore della costante C dipende dall'hardware utilizzato. Tale formula vale sia per il programma Pascal (fatto girare su C/128, su IBM, su Olivetti, poco importa: naturalmente il valore di C per il C/128 è molto più grande), sia per quello Basic (Basic Commodore su C/64, C/128 slow, C/128 fast; oppure MBasic su C/128-CP/M); e la formula continua a valere, con un "C" più piccolo, anche per il programma ottenibile compilando la versione Basic, ma con Austrospeed; il Petspeed maltratta i programmi ricorsivi.

Forme di visualizzazione diverse, che evitino i continui scrolling del video, potrebbero sensibilmente abbassare il valore della costante C; e d'altronde avevamo già osservato che, piuttosto che l'elenco delle mosse da effettuare, sarebbe più piacevole ottenere la visualizzazione sullo schermo, mossa dopo mossa, della distribuzione dei dischi sulle tre torri.

Se il numero di dischi non è elevato (8 al massimo) in Basic la soluzione più semplice è offerta dall'uso degli sprite. Per spostare uno sprite è sufficiente sapere quale è il suo numero, e da dove a dove spostarlo; una diversa visualizzazione (ad esempio consistente nel disegnare, mossa dopo mossa, il profilo delle tre torri) richiederebbe un rimaneggiamento del programma: non basta più sapere quale è il disco da spostare, ma serve anche sapere quali altri

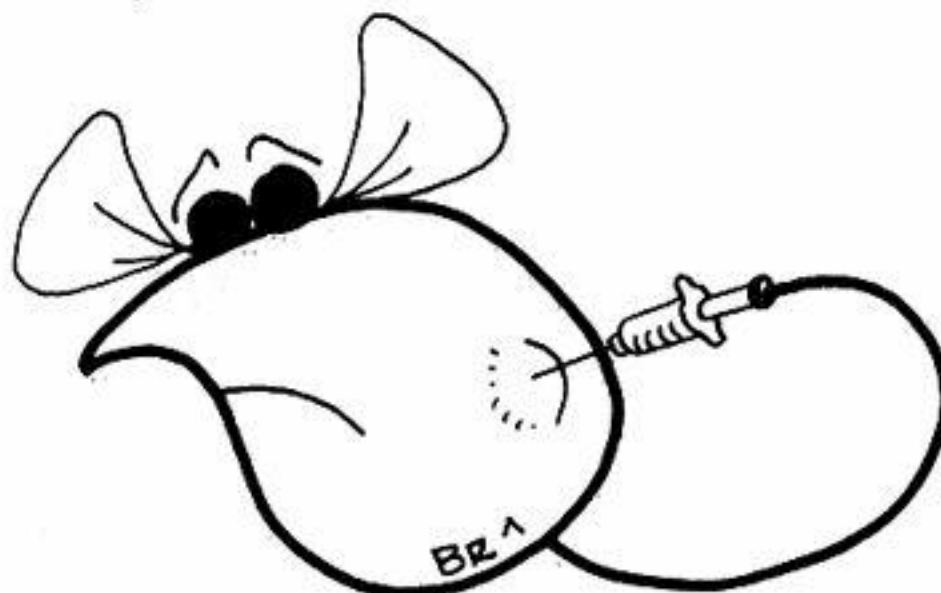
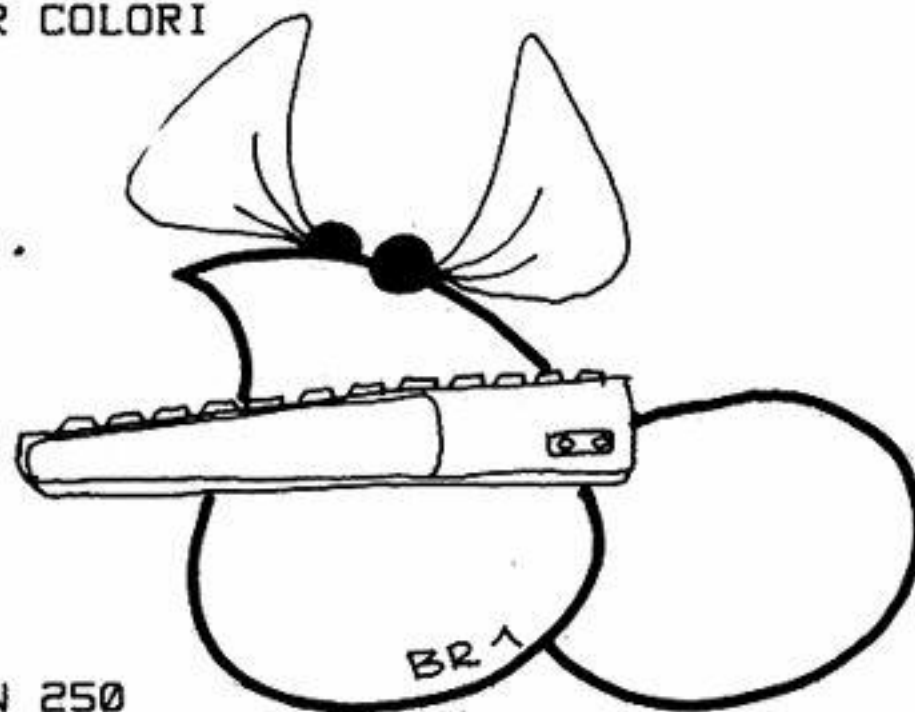
dischi sono presenti sulle torri tra cui ci si muove.

Si tratta, forse, di un utile esercizio di programmazione, che lasciamo senz'altro ai lettori limitandoci ad esplicitare il nocciolo del problema: quale è la "struttura" più adatta per tale memorizzazione? Ad esempio in Basic si potrebbe pensare a delle variabili di tipo stringa, o a degli array; cosa conviene di più?

In Pascal il problema è molto più vasto, perchè altre strutture (quali ad esempio quella di "insieme" e quella di "variabile dinamica") potrebbero a priori fornire soluzioni valide. E d'altronde, una volta deciso di scrivere un programma che, passo passo, "aggiorna" la situazione delle tre torri, si potrebbe anche pensare di seguire strade diverse da quella del promemoria, non necessariamente ricorsive.

Naturalmente la soluzione sprite evita queste difficoltà, ma sembra impraticabile in Turbo Pascal (chiunque disponesse, tuttavia, di notizie relative all'uso sotto CP/M di sprite, suoni, grafica hi-res eccetera è VIVAMENTE pregato di trasmettere tali informazioni all'autore).

Terminiamo proponendo un problema un po' più complicato: torniamo nei panni del Monaco Istruttore e supponiamo che si presenti da noi un allievo che, in seguito ad un colpo di sole, ha interrotto il lavoro e non ricorda più cosa stava facendo: i dischi sono distribuiti sulle tre torri (su ognuna rispettando l'ordine corretto) ma si sa solo quale è la torre finale; cosa facciamo per portare a termine le operazioni nel minor numero di mosse? C'è tempo per pensarci, magari ne riparleremo prossimamente.

```
5 REM      SOLUZIONE DEL PROBLEMA DELLE TORRI DI HANOI
6 REM COMPUTER: C/128 MODO 128, 80 COLONNE, MONITOR COLORI
7 REM                    BY CLAUDIO BAIOCCHI
8 :
10 DIMU,V,H,NM,A,B,C,D,X,Y,D$(15),T(3,25),RIGA(3),COLO(3)
60 SP$-"                              ":REM 30 SPAZI
80 C$-"[illegible]":REM COLORI DA BIANCO IN POI
110 COLO(1)-15:COLO(2)-40:COLO(3)-64
120 I$-CHR$(233):F$-CHR$(223):D$(1)-I$
130 FORX-2TO15:D$(X)=D$(X-1)+"  ":NEXT
150 FORX-1TO15:D$(X)-MID$(C$,X,1)+D$(X)+F$:NEXT
180 FORY-1TO16:T(1,Y)=Y:NEXT
190 NM=0:INPUT"QUANTI DISCHI";ND:IFND>15THENEND
200 TI$="000000":RIGA(1)=1:RIGA(2)=ND+8:RIGA(3)=ND+1
210 T(2,RIGA(2))-16:T(3,RIGA(3))-16
220 A=1:B=3-(ND AND1):C=6-A-B
230 SCNCLR:PRINTCHR$(142);
240 FORX=1TOND:CHAR1,COLO(1)-X,X,D$(X),1:NEXT
250 CHAR1,COLO(1)-4,ND+2,"[illegible]PARTENZA"
260 CHAR1,COLO(2)-6,ND+9,"DESTINAZIONE
270 CHAR1,COLO(3)-4,ND+2,"TRANSITO
280 DO:GOSUB310:IFRIGA(2)=8THENEXIT
290 GOSUB330:LOOP
300 PRINT"[illegible]FATTO IN "NM" MOSSE":PRINTTI" JIFFIES":END
310 H=RIGA(A):RIGA(A)=H+1:CHAR1,COLO(A)-1,H,"  ":H=RIGA(B)-1
320 CHAR1,COLO(B)-1,H,D$(1),1:RIGA(B)=H:NM=NM+1:RETURN
330 IFT(A,RIGA(A))<T(C,RIGA(C))THENU=A:V=C:ELSEV=A:U=C
340 H=RIGA(U):RIGA(U)=H+1:N=T(U,H)
350 CHAR1,COLO(U)-N,H,LEFT$(SP$,N+N):H=RIGA(V)-1
360 CHAR1,COLO(V)-N,H,D$(N),1:T(V,H)=N:RIGA(V)=H
370 NM=NM+1:D=A:A=B:B=C:C=D:RETURN
375 REM INSERIRE, TRA LE RIGHE 360 E 370, EVENTUALE
376 REM ROUTINE DI ATTESA PRESSIONE TASTO
380 END
```

```
10 REM SOLUZIONE DEL PROBLEMA DELLE TORRI DI HANOI
20 REM C/128, MODO 128, 40 COLONNE, MONITOR COLORI
30 REM          ANIMAZIONE MEDIANTE SPRITE
40 :
100 GOSUB 410 : REM INIZIALIZZAZIONE
110 GOSUB 540 : REM INPUT DATI, STRATEGIA
120 NM=0:GOSUB 220 : REM ESECUZIONE
130 GOSUB 680:REM SUONO
140 PRINT "FATTO IN" NM "MOSSE"
150 A$="S":INPUT "ANCORA"; A$
160 FOR X=1 TO 8:SPRITE X,0:NEXT
170 IF A$>"NO" THEN 110
180 GRAPHIC 0,1:END
190 :
200 REM ESECUZIONE
220 FOR X=1 TO ND:P=P(X):D=D(X):IF D=P THEN 250
230 N=X:GOSUB 360:REM SPOSTAMENTO SINGOLO
240 N=N-1:IF N THEN P=3-P-D:GOSUB 290:REM SPOSTA BLOCCO
250 NEXT:RETURN
260 :
270 REM ROUTINE RICORSIVA
290 IF N>1 THEN D=3-P-D:N=N-1:GOSUB 290:N=N+1:D=3-D-P
300 GOSUB 360:REM VISUALIZZAZIONE
310 IF N>1 THEN P=3-D-P:N=N-1:GOSUB 290:N=N+1:P=3-P-D
320 RETURN
330 :
340 REM VISUALIZZAZIONE MOSSE
360 X0=X(P):Y0=Y(P):X1=X(D):Y1=Y(D):FOR T=0 TO 1 STEP .125
370 MOVSPR N,X0+(X1-X0)*T,Y0+(Y1-Y0)*T:NEXT:NM=NM+1:RETURN
380 :
390 REM INIZIALIZZAZIONE
410 GRAPHIC 5,1
430 PRINT CHR$(15);:CHAR 1,33,13,"CAMBIA SCHERMO "
440 COLOR 0,15:COLOR 1,1:COLOR 5,2:GRAPHIC 1,1
450 FOR R=0 TO 2:READ X(R),Y(R):NEXT:DATA 200,120,50,120,125,50
460 FOR R=1 TO 8:CIRCLE 1,10,10,R+1:LOCATE 10,10:PAINT
470 SSHAPE A$,0,0,23,20:SPRSAV A$,R:SPRITE R,0,R,0,1,1,0:NEXT
480 GRAPHIC 2,1:CHAR 1,20,1,"DESTINAZIONE"
490 CHAR 1,1,16,"TRANSITO":CHAR 1,25,16,"PARTENZA"
500 RETURN
510 :
520 REM INPUT DATI, STRATEGIA
540 INPUT "■QUANTI DISCHI";ND:IF ND<1 OR ND>8 THEN 540
550 IF ND<1 OR ND>8 THEN GOSUB 680:GOTO 540
560 D=2:NM=0:FOR X=ND TO 1 STEP -1
570 PRINT "DOVE STA IL DISCO #" X " ? (P/T/D)";
580 DO:GETKEY A$:P=INSTR("PTD",A$)-1
590 IF P=-1 THEN GOSUB 680
595 LOOP UNTIL P>-1
600 PRINT A$:D(X)=D:P(X)=P
610 IF P<>D THEN NM=NM+2↑(X-1):D=3-P-D
620 MOVSPR X,X(P),Y(P):SPRITE X,1
630 NEXT:PRINT:PRINT:PRINT
640 PRINT "SERVIRANNO" NM " MOSSE":RETURN
650 :
660 REM SUONO
680 SOUND3,700,30:RETURN
690 END
```

CAMPUS

LABORATORIO SOFTWARE DI COMMODORE COMPUTER CLUB

CERCA COLLABORATORI

esclusivamente esterni (part - time) in possesso dei seguenti sistemi:

- **AMIGA** (modelli 500, oppure 2000, doppio drive, monitor a colori stereo, stampante).

- **MS/DOS** (Commodore PC o compatibili, dotati di hard disk, monitor a colori, stampante).

IL COLLABORATORE IDEALE:

- è uno studente universitario iscritto ad una facoltà tecnico - scientifica (o possiede cultura equivalente).
- ha sviluppato, in precedenza, notevole esperienza sul Commodore 64 oppure C/128.
- programma correntemente in Basic, Pascal, Assembly e/o C.
- risiede nell'hinterland milanese.
- è in grado di sviluppare autonomamente software di vario genere sulle macchine citate.

Gli interessati, previa telefonata, sosterranno un colloquio (nella sede di Viale Famagosta, in Milano) con l'ing. Alessandro de Simone, direttore di Commodore Computer Club, allo scopo di verificare l'effettiva possibilità di collaborazione.

Per informazioni: **Systems Editoriale**
Viale Famagosta, 75
Milano
Tel. (02) 84.67.348 (Telefonare dalle 15.00 alle 18.00 dal lunedì al venerdì)

AIUTA IL TUO EDICOLANTE A SERVIRTI MEGLIO

Se vuoi essere sicuro di trovare ogni mese *Commodore Computer Club* nella tua edicola, dai questo coupon al tuo edicolante, oppure spediscilo direttamente alla Systems Editoriale - via Mosè, 18 - 20090 Opera (MI)

Senza alcun impegno da parte mia, riservatemi ogni mese una copia di ....

Nome ____________

Via ____________ Tel. ____________ Città ____________

EDICOLA: Nome: ____________

Via ____________ Tel. ____________ Città ____________

INCOMINCIO DA TRE

Il modo peggiore per imparare a programmare è quello di scrivere subito programmi complessi. Un po' per volta, invece...

di **Alessandro de Simone**

Poche righe sono sufficienti per ottenere le prime soddisfazioni

La numerazione delle righe Basic, si sa, può essere qualunque dal momento che un efficiente sistema di editing provvede ad inserire, ove necessario, le righe digitate.

Si sa, inoltre, che è bene che la numerazione abbia un "passo" di 10, in modo, appunto, da favorire eventuali, futuri inserimenti.

Molto spesso, tuttavia, risulta utile usare un passo di maggiori dimensioni (50, 100). Quando iniziamo a scrivere un programma, infatti, non è possibile sapere quante nuove righe risulterà necessario inserire. Gli esempi riportati in queste pagine tendono, appunto, a dimostrare quanto detto.

Digitate i due gruppi di tre programmi riportati in queste pagine seguendo accuratamente le istruzioni qui indicate. Vi accorgerete che, per passare da una versione alla successiva, è sufficiente aggiungere le righe che mancano.

Le versioni più ridotte, quindi, vanno digitate così come le vedete; in seguito capirete il motivo delle righe che contengono solo il carattere di doppio punto (:).

MONO COLORE

Il primo programma della prima serie (Mono / colore) è formato da una manciata di righe; di queste non tutte contengono istruzioni. Vi consigliamo di digitarle, comunque, così come le vedete.

Il programma serve a visualizzare all'infinito (fichè non si preme il tasto Run / Stop) una qualsiasi stringa, carattere dopo carattere. Alla fine la stringa stessa viene cancellata e la procedura riprende.

Lanciando il programma con il solito Run, compare la domanda (vedi riga 130) "Digita una stringa?". A questo punto, digitando ad esempio il vostro nome, questo verrà associato alla variabile stringa A$.

Da notare che se, al momento di rispondere alla domanda, premete il tasto Return senza digitare nulla, alla variabile A$ rimane associata l'eventuale stringa precedentemente definita. Siccome (riga 120) ad A$ è associato il messaggio "Commodore Computer Club", questo rimarrà immutato all'interno del computer.

A questo trucchetto si ricorre spesso, in ambiente Basic (ma non funziona nel Gw-Basic Microsoft), quando non si voglia costringere l'utente a digitare stringhe che, nella maggior parte dei casi, possono esser predefinite.

Se, ovviamente, si digita qualcosa (anche un solo carattere!) prima di premere il tasto Return, il contenuto precedente di A$ viene irrimediabilmente perso, sostituito dalla nuova stringa.

Nella riga 170, alla variabile L viene associata la lunghezza della stringa A$. In questo modo il computer, in seguito, sarà in grado di individuare, grazie all'istruzione Mid$, i vari caratteri da porre in successione.

Il ciclo For...Next (riga 250) è il cosiddetto ciclo di ritardo. Il computer, infatti, non fa altro che... contare da 1 a 10. Per far ciò, tuttavia, impiega un certo tempo durante il quale l'elaborazione, in pratica, si sospende. Modificando il valore 10 con altri valori, minori o maggiori, si otterrà, rispettivamente, una maggiore o minore velocità di visualizzazione.

Per una serie di motivi (che invitiamo a scoprire da soli) la stringa non può contenere più di 39 caratteri. I caratteri eventualmente digitati in sovrannumero verranno visualizzati in modo molto particolare...

Miglioramenti

Utilizzando l'istruzione Left$ fate in modo che la stringa da "trattare" sia sempre lunga, al massimo, 39 caratteri (If L>39 Then...).

I più evoluti possono fare in modo che, alla fine della visualizzazione completa, il computer non cancelli tutto lo schermo, ma solo ciò che ha visua-

LE AVVENTURE DI

PRIMO GIOVEDINI by Marco Mietta

Una storia di "pirati"...

I CARATTERI SPECIALI E L'AMBIENTE MS-DOS

Ricordiamo che i codici di alcuni caratteri speciali rappresentano un comando per il computer.
Ad esempio...
Print Chr$(65)
...visualizza la vocale "A" (carattere alfanumerico) mentre...
Print Chr$(147)
...rappresenta un comando (cancella lo schermo).

Nei programmi di queste pagine sono chiamati in causa diversi codici di carattere; alcuni sono caratteri semigrafici. Altri, invece, rappresentano particolari comandi che qui elenchiamo:

Chr$(18) :imposta il modo Reverse
Chr$(146) :annulla il modo reverse.
Chr$(147) :cancella lo schermo.
Chr$(19) :posiziona il cursore in alto a sinistra.
Chr$(29) :sposta il cursore a destra.
Chr$(157) :sposta il cursore a sinistra.
Chr$(145) :sposta il cursore in alto.
Chr$(17) :sposta il cursore in basso.

Nel Gw-Basic originale Microsoft alcuni di tali comandi sono assenti; gli altri, invece, hanno codice diverso dal Basic Commodore. Si esmini l'apposito listato Gw-Basic per rendersene conto.

lizzato (sovrapponendovi, ad esempio, una stringa B$ formata da tanti caratteri di spazio Chr$(32) quanti sono i caratteri di A$).

I più bravi, infine, possono fare in modo da evitare la strana elaborazione che si verifica con L maggiore di 39.

```
100 REM PRIMA FASE: MONO/COLORE
110 :
120 A$="COMMODORE COMPUTER CLUB"
130 INPUT "DIGITA UNA STRINGA";A$
140 :
150 :
160 PRINT CHR$(147)
170 L=LEN (A$)
180 :
190 FOR X=1 TO L
200 :
210 PRINT TAB(X);
220 :
230 :
240 PRINT MID$(A$,X,1); CHR$(145)
250 FOR K=1 TO 10: NEXT
260 NEXT
270 :                    GOTO 160
```

BI-COLORE

Il secondo programma (Bi / Colore) è praticamente identico al precedente; in alcune righe "vuote" di quest'ultimo, infatti, sono state aggiunte altre istruzioni. Vi consigliamo, quindi, di non digitarlo per intero, ma di aggiungere, al programma "Mono Colore", solo le istruzioni che occorrono affinchè diventi identico al secondo listato.

Quest'ultimo visualizza ancora una stringa alfanumerica (lunga non più di 39 caratteri) ma, alternativamente, in modo normale ed in reverse.

In riga 170 è stata aggiunta una variabile numerica (W=0) che servirà come "deviatore". Il computer, infatti, dovrà pur sapere, in qualche modo, quando è il caso di visualizzare un messaggio in "normale" oppure in reverse.

Ebbene, il trucchetto consiste in questo: non appena il messaggio viene visualizzato in "normale", la variabile W viene posta ad "1" (riga 260: If... W=1). Quando il ciclo riprende (goto 180), il programma (vedi riga 220) stamperà il carattere di codice 18 che, però, non è un vero e proprio carattere, ma rappresenta il comando: "attiva il modo reverse".

9 Febbraio 1989, TI$="124500": Primo Giovedini è in volo per una nuova missione con il suo F18.
Egli non sa ancora cosa lo attende: infatti deve giungere nel settore previsto ed aprire la busta di istruzioni...

```
100 REM SECONDA FASE: BI/COLORE
110 :
120 A$="COMMODORE COMPUTER CLUB"
130 INPUT "DIGITA UNA STRINGA";A$
140 :
150 :
160 PRINT CHR$(147)
170 L=LEN (A$): Q$=CHR$(18): W=0
180 :
190 FOR X=1 TO L
200 :
210 PRINT TAB(X);
220 IF W=1 THEN PRINT Q$;
230 :
240 PRINT MID$(A$,X,1); CHR$(145)
250 FOR K=1 TO 10: NEXT
260 NEXT: IF W=0 THEN W=1:GOTO 180
270 IF W=1 THEN W=0: GOTO 180
```

Il messaggio viene quindi riportato nero su bianco (notare il carattere di punto e virgola presente dopo Q$ in riga 220). Poichè, ora, la variabile W vale uno (1), la parte terminale della riga 260 non verrà eseguita. La riga 270, invece, provvederà ad annullare la variabile W che, al ciclo successivo, farà in modo di "evitare" le istruzioni di riga 220 ed a visualizzare il messaggio in modo "normale".

Miglioramenti

Inserite un altro deviatore in modo che il messaggio appaia, oltre che in bianco ed in bianco-reverse, anche in nero ed in nero-reverse.

DUE OPZIONI

Quest'ultimo programma mini-serio della mini-serie (freddura squallida) è l'ultima "evoluzione" che suggeriamo al lettore principiante. Anche questo listato, come i primi due, visualizza un messaggio. Stavolta, però, è possibile stabilire se la conversione normale - reverse (e viceversa) debba avvenire da sinistra a destra oppure al contrario.

Si noti che è possibile ottenere il listatao apportando le modifiche necessarie al secondo programma. Anche in questo caso, infatti, i due listati sono totalmente compatibili, nel senso che uno è la "derivazione" dell'altro.

Anche in questo caso è stato utilizzato un deviatore (variabile numerica R) che viene automaticamente posto a 1 oppure a 0 dopo le opportune elaborazioni.

Stavolta la procedura è un po' più complessa: è infatti necessario effettuare un ciclo For...Next con step positivo (implicito: riga 190) ed uno con step esplicitamente negativo (riga 200).

L'attività incrociata dei due deviatori ("R" e "W") si presenta laboriosa, ma non troppo difficile da capire; almeno per i lettori più volenterosi.

Miglioramenti

Fate in modo che il messaggio appaia, alternativamente, da destra a sinistra e da sinistra a destra.

Introducete, poi, un'opzione con cui impostare il modo "reverse" e "normale", carattere dopo carattere. Buon lavoro!

UNA PALLINA VAGANTE

La seconda serie di mini listati provvede a creare un'animazione in cui una pallina vaga per lo schermo in modo casuale.

Anche in questo caso vi consigliamo di digitare i tre programmi partendo dal primo. Questo, apportandovi alcune modifiche, diventerà identico al secondo; quest'ultimo, in seguito, assumerà l'aspetto del terzo. La procedura serve per dimostrare, ancora una volta, che è possibile sofisticare i propri programmi a patto di iniziare la loro stesura tenendo conto di ciò che, in seguito, possiamo desiderare.

Il primo programma definisce (riga 120) il carattere semi grafico di codice 215 che è, appunto, una pallina. Volendo, potete sostituire il codice che più

```
100 REM TERZA FASE: DUE OPZIONI
110 :
120 A$="COMMODORE COMPUTER CLUB"
130 INPUT "DIGITA UNA STRINGA";A$
140 INPUT "DESTRA O SINISTRA (1/2)";R$
150 R=1: IF R$<>"1" THEN R=0
160 PRINT CHR$(147)
170 L=LEN (A$): Q$=CHR$(18): W=0
180 IF R=0 THEN 200
190 FOR X=1 TO L: GOTO 210
200 FOR X=L+1 TO 1 STEP -1
210 PRINT TAB(X);
220 IF W=1 THEN PRINT Q$;
230 IF R=0 THEN PRINT CHR$(32);
240 PRINT MID$(A$,X,1); CHR$(145)
250 FOR K=1 TO 10: NEXT
260 NEXT: IF W=0 THEN W=1:GOTO 180
270 IF W=1 THEN W=0: GOTO 180
```

Ricordatevi di premere sempre il tasto Return alla fine di ciascun rigo digitato

PRIMA E DOPO

La volta scorsa abbiamo esaminato il modo di usare alcune istruzioni dei piccoli computer Commodore (C/64-128, C/16) suggerendo diverse applicazioni pratiche.

Stavolta, continuando il discorso "didattico", esamineremo, attraverso due esempi di tre "fasi" ciascuno, come sia possibile raggiungere risultati interessanti partendo da programmi di dimensioni minime.

E' ovvio che metteremo in evidenza la procedura generale; al lettore (principiante) è affidato il compito di assimilare la "sostanza" in modo da proseguire con i propri passi.

Ricordiamo che, per raggiungere risultati discreti, è indispensabile leggere con attenzione il libretto di istruzioni del proprio computer. Forse (specialmente nel caso del C/128) le sue dimensioni possono apparire scoraggianti, e gli argomenti trattati, non tanto chiari. E' bene, però, che il lettore si abitui ad apprendere ciò che serve evitando, a proprie spese, di fare la scoperta dell'acqua calda: confusione di zero (0) con la vocale O, mancata pressione del tasto Return quando necessario, digitazione di comandi tipo Poke e Sys con troppa disinvoltura e così via; pericoli, questi, dettagliatamente indicati nel famoso e bistrattato libretto di istruzioni.

Provate ad inserire i numerosi codici di comando disponibili sul C/64

vi aggrada (provate con 209, 211 e così via).

Subito dopo (righe 150 - 180) alle stringhe A$, B$, C$, D$ vengono associati i caratteri di comando che impongono al cursore, rispettivamente, di dirigersi in basso, a sinistra, in alto e a destra.

La variabile DL (riga 190) ha il compito di determinare il ciclo di ritardo (riga 210): con DL = 0 l'animazione sarà rapidissima, quasi fastidiosa a vedersi; con valori maggiori è possibile rallentare il movimento della pallina.

La riga 240 estrae un numero casuale, compreso tra 1 e 4, in modo che (riga 260) venga attivata una delle quattro righe (270, 280, 290, 300) che sovrintendono al movimento della pallina stessa.

Alla variabile Y (riga 220 e For..Next delle righe 270... 300) è affidato il compito di ripetere per Y volte il movimento sorteggiato casualmente. Con Y=10, ad esempio, la pallina si muoverà per 10 volte nella direzione che, volta per volta, verrà sorteggiata, prima di scegliere una nuova direzione casuale.

La stringa L1$ (riga 130) è formata da tre caratteri: cursore a sinistra, spazio e di nuovo cursore a sinistra. Tale stringa, di notevole importanza, serve sia a cancellare il carattere appena visualizzato, sia a posizionare il cursore nella stessa posizione di prima. In assenza di tale stringa (provate a cencellare la riga 130) il carattere stesso lascerebbe una scia durante i suoi spostamenti.

E' bene sottolineare l'importanza del carattere di punto e virgola (;) presente in numerosissime righe del listato. Ricordiamo che tale segno di punteggiatura (applicabile solo in caso di istruzioni Print) impone al computer di visualizzare l'eventuale, successivo carattere, subito dopo la presenza dell'ultimo carattere visualizzato. In mancanza del punto e virgola, insomma, i caratteri verrebbero sempre stampati all'inizio del rigo successivo.

Miglioramenti

Fate in modo che, dopo un certo intervallo di tempo, la velocità aumenti sempre di più ed il carattere visualizzato cambi aspetto. Fate in modo, inoltre, che il carattere lasci la scia, o la cancelli, in modo casuale.

PALLINA PRINGIONIERA

Il secondo listato, ricavabile dal primo apportandovi minime modifiche, elimina un inconveniente.

Nel suo girovagare, infatti, la pallina del primo programma presentava un'anomalia. Giunta all'estrema sinistra (o all'estrema destra) riappariva nell'ultima colonna di destra (nella prima di sinistra) provocando un'incongruenza nell'animazione stessa.

Ciò è dovuto al fatto che per il computer non esiste un vero e proprio confine dal momento che, per lui, lo schermo non è altro che un'unica lunga striscia di 1000 quadratini (1000 = 25 righe x 40 colonne).

Il programma n.2, invece, riesce a stabilire se è

stato raggiunto uno dei due limiti estremi, destro o sinistro, ed impedisce, in uno di questi due casi, che l'animazione della pallina provochi l'incongruenza accennata.

Per raggiungere tale risultato è sufficiente tener conto del numero di "passi" compiuti dalla pallina a destra ed a sinistra, e comportarsi di conseguenza.

Il problema non si presenta, invece, quando il movimento è verso l'alto o verso il basso. Nel primo caso, infatti, la pallina non procede; nel secondo viene attivato lo scroll (scorrimento) e la simulazione non risente di "scatti" improvvisi.

Ricordiamo che viene attivato il movimento a destra e a sinistra se viene estratto il numero casuale (riga 240) 4 oppure 1. In questi casi, e solo in questi casi (righe 240, 250: If...), si salta alla subroutine 330 (movimento a destra) oppure 350 (sinistra) per vedere se è il caso di eseguire l'ordine. Qui sono memorizzate le posizioni del cursore: se questo, in seguito all'eventuale spostamento, dovesse trovarsi "oltre" lo schermo, il movimento viene interdetto. A ciò provvede la variabile A che, funzionando come deviatore, impone (If A=1... righe 240 e 250) di estrarre un nuovo numero casuale.

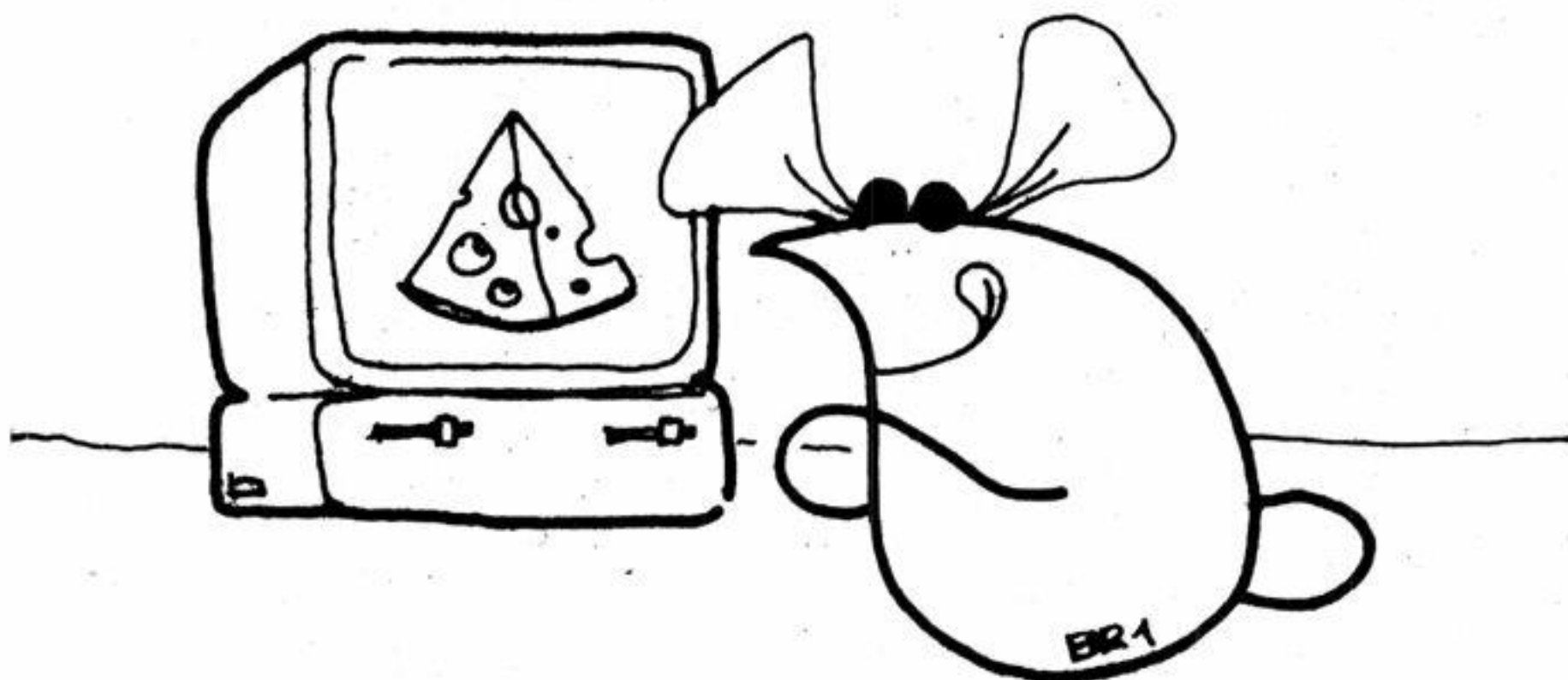
La variabile U, invece, memorizza, spostamento dopo spostamento, l'effettiva posizione della pallina.

In riga 220 lo spostamento, misurato in numero di caselle da percorrere, viene determinato casualmente (variabile Y).

Miglioramenti

Fate in modo che la pallina, tutte le volte che "urta"

In ambiente Gw-Basic è necessario apportare alcune modifiche ai listati pubblicati

```
100 REM PRIMA FASE: PALLINA VAGANTE
110 :
120 P$=CHR$(215):REM PALLINA
130 L1$=CHR$(157)+CHR$(32)+CHR$(157)
140 :
150 A$=CHR$(17): REM BASSO
160 B$=CHR$(157):REM SINISTRA
170 C$=CHR$(145):REM ALTO
180 D$=CHR$(29): REM DESTRA
190 :                          : DL=50
200 :
210 :
220 Y=3
230 :
240 X=INT(RND(0)*5)
250 :
260 ON X GOSUB 270,280,290,300:GOTO 220
270 FOR J=1 TO Y:PRINT A$;P$;:GOSUB 310:NEXT:RETURN
280 FOR J=1 TO Y:PRINT B$;P$;:GOSUB 310:NEXT:RETURN
290 FOR J=1 TO Y:PRINTC$;P$;:GOSUB310:NEXT:RETURN
300 FOR J=1 TO Y:PRINTD$;P$;:GOSUB310:NEXT:RETURN
310 FOR I=1 TO DL: NEXT:          PRINT L1$;:RETURN
320 :
330 :
340 :
350 :
360 :
370 END
```

```
100 REM SECONDA FASE: PALLINA PRIGIONIERA
110 :
120 P$=CHR$(215):REM PALLINA
130 L1$=CHR$(157)+CHR$(32)+CHR$(157)
140 :
150 A$=CHR$(17): REM BASSO
160 B$=CHR$(157):REM SINISTRA
170 C$=CHR$(145):REM ALTO
180 D$=CHR$(29): REM DESTRA
190 :                          : DL=50
200 :
210 :
220 Y=INT(RND(0)*18):REM ESCURSIONE
230 :
240 X=INT(RND(0)*5):IF X=4 THEN GOSUB 330:IF A=1 THEN 240
250 IF X=2 THEN GOSUB 350:IF A=1 THEN 240
260 ON X GOSUB 270,280,290,300:GOTO 220
270 FOR J=1 TO Y:PRINT A$;P$;:GOSUB 310:NEXT:RETURN
280 FOR J=1 TO Y:PRINT B$;P$;:GOSUB 310:NEXT:RETURN
290 FOR J=1 TO Y:PRINTC$;P$;:GOSUB310:NEXT:RETURN
300 FOR J=1 TO Y:PRINTD$;P$;:GOSUB310:NEXT:RETURN
310 FOR I=1 TO DL: NEXT:              PRINT L1$;:RETURN
320 :
330 A=0:U=U+Y: IF U>39-Y  THEN U=U-Y:A=1
340 RETURN
350 A=0:U=U-Y: IF U<2 THEN U=U+Y:A=1
360 RETURN
370 END
```

```
100 REM TERZA FASE: PALLINA CON SCIA
110 :
120 P$=CHR$(215):REM PALLINA
130 L1$=CHR$(157)+CHR$(32)+CHR$(157):REM NORMALE
140 L2$=CHR$(157)+CHR$(18)+CHR$(32)+CHR$(157)+CHR$(146):REM REVERSE
150 A$=CHR$(17): REM BASSO
160 B$=CHR$(157):REM SINISTRA
170 C$=CHR$(145):REM ALTO
180 D$=CHR$(29): REM DESTRA
190 PRINT CHR$(147);CHR$(18);: DL=50
200 FOR I=1TO 999:PRINTCHR$(32);:NEXT
210 PRINT CHR$(19)CHR$(146);
220 Y=INT(RND(0)*18):REM ESCURSIONE
230 Z=INT(RND(0)*10):REM REVERSE ON/OFF
240 X=INT(RND(0)*5):IF X=4 THEN GOSUB 330:IF A=1 THEN 240
250 IF X=2 THEN GOSUB 350:IF A=1 THEN 240
260 ON X GOSUB 270,280,290,300:GOTO 220
270 FOR J=1 TO Y:PRINT A$;P$;:GOSUB 310:NEXT:RETURN
280 FOR J=1 TO Y:PRINT B$;P$;:GOSUB 310:NEXT:RETURN
290 FOR J=1 TO Y:PRINTC$;P$;:GOSUB310:NEXT:RETURN
300 FOR J=1 TO Y:PRINTD$;P$;:GOSUB310:NEXT:RETURN
310 FOR I=1 TO DL: NEXT:IF Z<5 THEN PRINT L1$;:RETURN
320 PRINT L2$;:RETURN
330 A=0:U=U+Y: IF U>39-Y  THEN U=U-Y:A=1
340 RETURN
350 A=0:U=U-Y: IF U<2 THEN U=U+Y:A=1
360 RETURN
370 END
```

contro il limite destro o sinistro, cambi forma (If A=1 Then...). Fate in modo, poi, da determinare la "fascia" verticale entro la quale è possibile il movimento della pallina stessa.

L'ULTIMO LISTATO, ANCHE IN GW-BASIC

L'ultimo programma, sul quale ci intratteniamo pochissimo, è ricavabile dal secondo apportando le dovute aggiunte.

Grazie all'uso intensivo di deviatori e di caratteri speciali la pallina si muove all'interno di una fascia verticale ben definita e lascia una scia, oppure no, in modo casuale.

I "disegni" che è possibile ottenere sono infiniti, soprattutto inserendo ulteriori modifiche atte a determinare casualmente la velocità di animazione, la modifica del carattere, la quantità di caselle da percorrere.

Chi ci ha seguito fin qui, pertanto, sarà sicuramente i grado di effettuare le modifiche che più desidera.

Viene fornita, come accennato all'inizio, anche la versione Gw-Basic Microsift, sia per accontentare gli utenti Commodore in possesso di un PC, sia per evidenziare le differenze esistenti tra i due tipi di computer.

```
100 REM Gw-Basic per computer MS-DOS
110 REM pallina con scia
120 P$=CHR$(64):REM pallina
130 L1$=CHR$(29)+CHR$(32)+CHR$(29 )
140 KEY OFF
150 A$=CHR$(31): REM basso
160 B$=CHR$(29): REM sinistra
170 C$=CHR$(30): REM alto
180 D$=CHR$(28): REM destra
190 PRINT CHR$(12 );:DL=50
200 :
210 :
220 Y=INT(RND(1)*18):REM escursione
230 Z=INT(RND(1)*10):REM reverse on/off
240 X=INT(RND(1)*5):IF X=4 THEN GOSUB 330:IF A=1 THEN 240
250 IF X=2 THEN GOSUB 350:IF A=1 THEN 240
255 IF Z<5 THEN COLOR 0,7:ELSE:COLOR 1,1
260 ON X GOSUB 270,280,290,300
261 GOTO 220
270 FOR J=1 TO Y:PRINT A$;P$;:GOSUB 310:NEXT:RETURN
280 FOR J=1 TO Y:PRINT B$;P$;:GOSUB 310:NEXT:RETURN
290 FOR J=1 TO Y:PRINT C$;P$;:GOSUB 310:NEXT:RETURN
300 FOR J=1 TO Y:PRINT D$;P$;:GOSUB 310:NEXT:RETURN
310 FOR I=1 TO DL: NEXT
320 PRINT L1$;:RETURN
330 A=0:U=U+Y: IF U>79-Y  THEN U=U-Y:A=1
340 RETURN
350 A=0:U=U-Y: IF U<2 THEN U=U+Y:A=1
360 RETURN
370 END
```

OLTRE LA MUSICA DEL SID

Come costringere il C/128 (ed il C/64) a "trattare" i segnali audio provenienti dall'esterno

di **Moreno Paisi**

Un semplice collegamento h/w consente di ampliare le potenzialità del C/64-128

Attraverso il piedino 5 del connettore Audio / Video del C/64 è possibile immettere, nel computer, un segnale audio di bassa frequenza, come quello proveniente da un impianto Hi-Fi (relativo ad un solo canale), da un videoregistratore, da un microfono o da una qualsiasi sorgente di segnale ad alta impedenza, (la resistena di ingresso è di circa 100 kohm), a condizione di non superare determinati valori di ampiezza del segnale in ingresso.

Il segnale applicato verrà diffuso immediatamente dall'altoparlante del televisore (o del monitor) collegato al computer, senza bisogno di software specifico. Potrà quindi essere ascoltato così come viene immesso oppure, ricorrendo ad apposito software, elaborandolo sfruttando il filtro interno del computer.

Questa particolarità permette a tutti gli utenti di C/64-128 di sfuttare quella che è la meno conosciuta delle caratteristiche di questi computer, e che si riscontra quasi prevalentemente in apparecchiature di categoria superiore.

Occorre precisare, ad ogni buon conto, che le operazioni qui di seguito descritte, anche se molto semplici, possono portare al danneggiamento del computer o della sorgente esterna, se effettuate in modo errato. Consigliamo, quindi, di realizzarle solo se in possesso di un'adeguata preparazione al riguardo. La lettura del presente articolo, in ogni caso, vi permetterà di conoscere meglio l'hardware del vostro calcolatore.

Chi possiede lo schema elettrico del computer ("Guida di riferimento per il programmatore") può notare che il piedino 5 del connettore A/V, attraverso un condensatore, è connesso al piedino 26 del circuito integrato MOS 6581, meglio conosciuto come SID (Sound Interface Device).

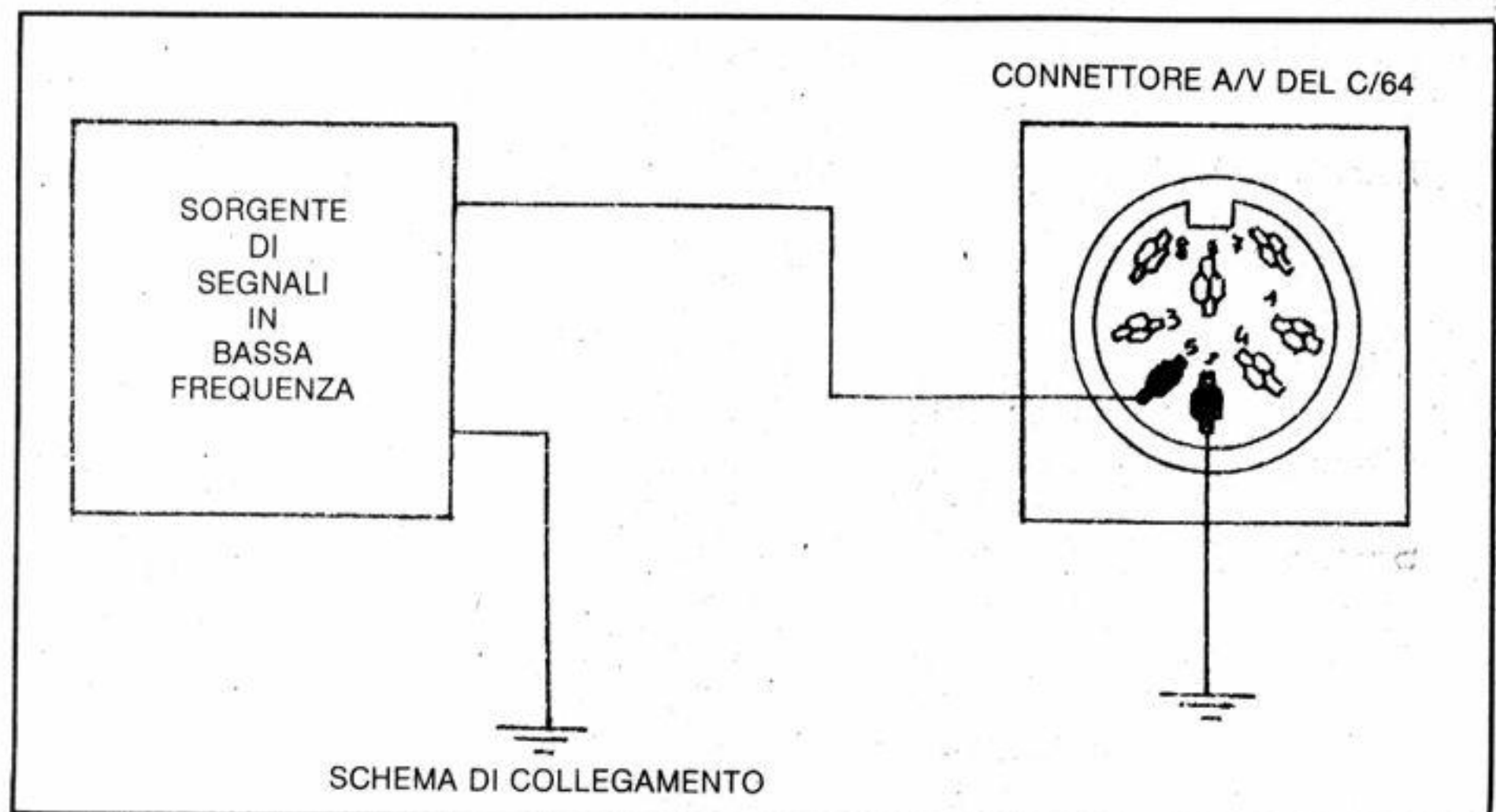
EL SID

Il SID è il dispositivo che gestisce le caratteristiche sonore del C/64 e rende disponibili le seguenti funzioni:

- Generazione di quattro forme d'onda: triangolare, dente di sega, impulso variabile e rumore bianco.
- Possibilità di modulare il segnale generato.
- Controllo del volume principale: viene gestito via software dal comando VolXX nel basic 7.0 (C/128) oppure tramite l'uso di Poke 54296, XX nel basic 2.0 del C/64; XX può assumere valori compresi tra 0 e 15.
- Filtro programmabile: il filtro è in grado di filtrare il suono in tre modi: passa-basso, passa-banda e passa-alto. Questi tipi di filtro possono essere usati singolarmente oppure addizionandoli, ottenendo filtri di tipo passa-basso + passa-banda, passa-banda + passa-alto e passa-basso + passa-alto.
- Dispone di due interfacce analogico - digitali, in grado di trasformare un segnale analogico in digitale, denominate Pot X e Pot Y, controllate da un paddle inserito nelle porte del controller.
- Ingresso audio esterno.

Il SID dispone di tre voci indipendenti, che possono essere usate singolarmente, combinate tra loro seppure con fonti audio esterne, da applicare all'ingresso Audio In.

Tramite quest'ultimo ingresso è possibile miscelare i segnali audio esterni con l'uscita del SID (così come farebbe un mixer) oppure passarli attraverso il filtro interno.

E' bene ricordare che, per salvaguardare l'incolumità del computer, è necessario che l'ampiezza del segnale d'ingresso non sia superiore ai 3 volt picco-picco.

```
MODIFICHE PER VERSIONE C/64

100 print chr$(147)
110 co=32: rg=5: gosub 440: print chr$(95) : a=0: b=0: c=0
150 co=10: rg=5: gosub 440: print"Filtro passa basso "+a$(a)
160 rg=8: gosub 440: print"Filtro passa banda "+a$(b)
170 rg=11: gosub 440: print"Filtro passa alto  "+a$(c)
180 rg=14: gosub 440: print"Volume ="+str$(vo)+" "
190 rg=17: gosub 440: print"Frequenza ="+str$(fr)+" "
200 rg=20: gosub 440: print"Risonanza ="+str$(ri)+" "
300 gosub 450
330 co=32: rg=z: print" ": rg=y: gosub 450:  print chr$(95): z=y
440 poke 211,co: poke 214,rg: sys 58640:  return
450 w=peek(56320): if w=127 then 450
460 if w<112 then j=128
470 if w=126 then j=1
480 if w=125 then j=5
490 if w=109 then j=133
500 if w=110 then j=129
510 return
```

```
LOCAZIONI UTILI PER GESTIRE IL FILTRO

Locaz. Bit  Descrizione
----------------------------------------

54293        Frequenza di taglio del
             filtro : parte bassa
54294        Frequenza di taglio del
             filtro : parte alta
54295  7-4   Selezione della risonanza
             del filtro : 0-15
         3   Ingresso esterno collegato
             filtro : 1 = SI
         2   Output voce 3 al filtro :
             1 = SI
         1   Output voce 2 al filtro :
             1 = SI
         0   Output voce 1 al filtro :
            1 = SI
54296    7   Outputdi taglio voce 3
         6   Seleziona il filtro
             passa-alto : 1 = SI
         5   Seleziona il filtro
             passa-banda : 1 = SI
         4   Seleziona il filtro
             passa-basso : 1 = SI
       3-0   Volume d'uscita : 0-15
```

Dopo aver effettuato i collegamenti, come mostrato in figura, ed averne controllato la corretta esecuzione, alzate a metà volume il televisore, accendete il computer e digitate...
VOL 15
...se si tratta di un C/128, oppure...
Poke 54296, 15
...se si tratta di un C/64. Dopo queste operazioni preliminari si dovrebbe udire il segnale introdotto; nel caso questo risultasse di livello basso, alzate il volume del televisore; se neanche quest'operazione dovesse apportare miglioramenti (e solo in questo caso) potete innalzare GRADUALMENTE il livello del segnale d'ingresso.

Non dovrebbe esser neccessario alcun componente di accoppiamento in quanto gli elementi utili sono già presenti all'interno del computer.

COME ELABORARE IL SEGNALE ESTERNO

Per utilizzare il filtro, allo scopo di elaborare il segnale introdotto, è neccessario ricorrere all'uso in-

Il progetto di queste pagine viene consigliato solo a chi se ne intende di elettronica

CHE COSA E' UN FILTRO

Un filtro è un particolare circuito elettronico in grado di separare segnali elettrici in funzione della loro frequenza.

Le caratteristiche dei quattro tipi fondamentali di filtro sono:

- Filtro passa-basso. Lascia passare solo i segnali di frequenza inferiore a quella di cut-off, detta anche frequenza di taglio, attenuando quelli di frequenza superiore.
- Filtro passa-alto. Agisce in modo contrario al precedente, lasciando passare solo i segnali superiori alla frequenza di taglio.
- Filtro passa-banda. Lascia passare solo i segnali la cui frequenza è compresa entro una determinata banda, attenuando tutti gli altri.
- Filtro ad esclusione di banda. Al contrario del precedente, questo filtro lascia passare tutti i segnali le cui frequenze non siano comprese entro una determinata banda. Questo filtro può essere realizzato combinando, tra loro, filtri di tipo passa-basso e passa-alto.

I programmi qui riportati sono semplici applicazioni generali

tensivo dell'istruzione Poke.

Ricordiamo che le locazioni utilizzate per gestire il filtro sono quelle riportate in tabella.

Per attivare il filtro occorre agire come segue:

- si pone a "1" il bit 3 della locazione 54295, che, selezionando una risonanza media, corrisponde al valore decimale di 136.
- si trascrive, nelle locazioni 54293 e 54294, la frequenza desiderata (nella solita forma di parte alta e parte bassa); ad esempio, per una frequenza di 1000 si depositano i valori:
parte alta = int (1000 / 256 =3)
parte bassa = 1000 - (Parte alta * 256) = 235
- Infine si seleziona il tipo di filtro ed il volume, da depositare nella locazione 54296; ad esempio, la selezione di un filtro passa alto e l'impostazione del massimo volume, corrisponderanno, in binario, al numero 1000 / 1111, cioè a 79 in decimale.

IL PROGRAMMA

Attraverso il programma è possibile scegliere, tramite l'uso del joystick, i tipi di filtri da attivare, la frequenza di taglio dei singoli filtri, la risonanza ed il volume del segnale d'uscita.

Per attivare (on) / disattivare (off) un filtro è sufficiente posizionare il cursore (rappresentato dalla freccetta), muovendo verso l'alto o verso il basso il joystick, in corrispondenza del filtro prescelto, e premere il tasto di fire.

Per aumentare (decrementare) il valore del volume, della frequenza e della risonanza, bisogna posizionare il cursore in corrispondenza della caratteristica da modificare, premere il tasto del joystick e quindi muovere la leva verso l'alto (per aumentare) o verso il basso (per diminuire) il valore corrispondente.

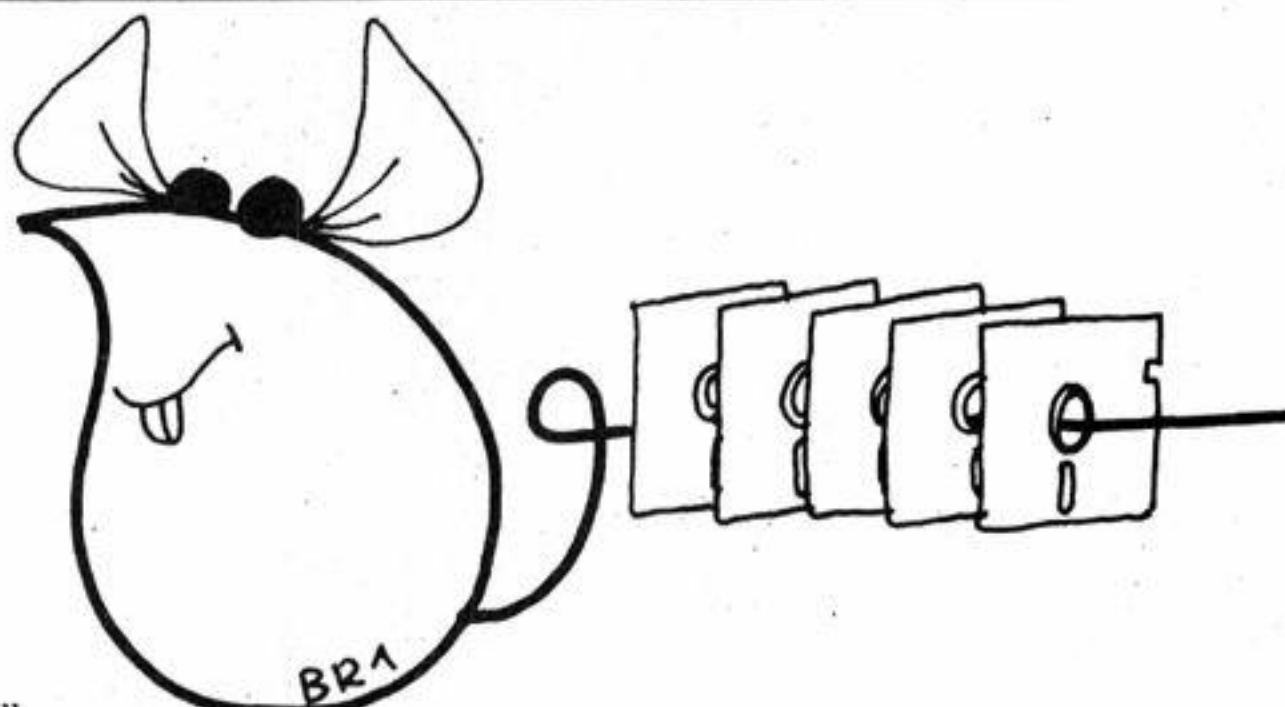
Per la frequenza e la risonanza non sarà udibile la variazione, in quanto avviene lentamente. I due parametri influiranno sul suono solo quando uno o più filtri saranno attivi.

Il programma riportato è adatto al solo C/128 a causa di alcune istruzioni Basic assenti nell'interprete del C/64.

MODIFICHE PER IL C/64

Per rendere compatibile il programma pubblicato con il C/64 bisogna innanzi tutto digitare il programma per il C/128, senza modificare la numerazione di riga, e quindi digitare le modifiche, riportate in queste pagine, che si sostituiranno automaticamente a quelle che contengono le istruzioni riservate al basic del C/128.

Le istruzioni, infatti, sono sostituite da altrettante routine che svolgono gli stessi compiti.

```
10 REM ************************
20 REM ***  SOUND  EFFECTS   ***
30 REM ***       BY          ***
40 REM ***  PAISI   MORENO   ***
50 REM ***     PER C/128     ***
60 REM ************************
90 :
100 SCNCLR:REM PULISCE LO SCHERMO
105 PRINT"(JOY IN PORTA 2)"
106 PRINT"SELEZIONARE CON FIRE (X)"
107 PRINT"SELEZIONARE CON FIRE + JOY UP/DOWN (Y)"
110 CHAR1,32,5,CHR$(95):A=0:B=0:C=0
120 Y=5:Z=5:VO=8:FR=2000:RI=8:A$(0)="OFF":A$(1)="ON ":GOTO 150:REM PARAM.INIZ.
130 GOSUB300
140 REM *** MASCHERA VIDEO ***
150 CHAR1,2,5,"(X)   FILTRO PASSA BASSO "+A$(A)
160 CHAR1,2,8,"(X)   FILTRO PASSA BANDA "+A$(B)
170 CHAR1,2,11,"(X)   FILTRO PASSA ALTO  "+A$(C)
180 CHAR1,2,14,"(Y)   VOLUME ="+STR$(VO)+" "
190 CHAR1,2,17,"(Y)   FREQUENZA ="+STR$(FR)+" "
200 CHAR1,2,20,"(Y)   RISONANZA ="+STR$(RI)+" "
210 IF A=0 AND B=0 AND C=0 THEN POKE 54295,0:POKE 54296,VO:GOTO 130:REM FILTRI E
SCLUSI
220 FA=INT(FR/256):POKE 54294,FA:REM FREQUENZA DI TAGLIO DEI FILTRI, PARTE ALTA
230 FB=FR-(FA*256):POKE 54293,FB:REM FREQUENZA DI TAGLIO DEI FILTRI, PARTE BASSA
240 T=A+(B*2)+(C*4)
250 POKE 54295,(16*RI)+8:POKE 54296,(16*T)+VO:GOTO 130:REM ATTIV.FILT.E RISON.
260 :
290 REM ROUTINE CONTROLLO JOYSTICK
300 J=JOY(2):IF J=0 THEN GOTO 300
310 IF J=1 THEN Y=Y-3:IF Y<5 THEN Y=5:REM CURSORE SU LINEA PRECEDENTE
320 IF J=5 THEN Y=Y+3:IF Y>20 THEN Y=20:REM CURSORE SU LINEA SUCCESSIVA
330 CHAR1,32,Z," ":CHAR 1,32,Y,CHR$(95):Z=Y
340 IF Y=5 AND J>127 THEN A=A+1:IF A>1 THEN A=0:REM CAMBIA STATO PASSA BASSO
350 IF Y=8 AND J>127 THEN B=B+1:IF B>1 THEN B=0:REM CAMBIA STATO PASSA BANDA
360 IF Y=11 AND J>127 THEN C=C+1:IF C>1 THEN C=0:REM CAMBIA STATO PASSA ALTO
370 IF Y=14 AND J=129 THEN VO=VO+1:IF VO>15 THEN VO=15:REM VOLUME+1
380 IF Y=14 AND J=133 THEN VO=VO-1:IF VO<0 THEN VO=0:REM VOLUME-1
390 IF Y=17 AND J=129 THEN FR=FR+50:IF FR>4000 THEN FR=4000:REM FREQUENZA+50
400 IF Y=17 AND J=133 THEN FR=FR-50:IF FR<0 THEN FR=0:REM FREQUENZA-50
410 IF Y=20 AND J=129 THEN RI=RI+1:IF RI>15 THEN RI=15:REM RISONANZA+1
420 IF Y=20 AND J=133 THEN RI=RI-1:IF RI<0 THEN RI=0:REM RISONANZA-1
430 RETURN
```

C/128, ED E' SUBITO RASTER

Come sfruttare, nella pratica, le piccole differenze con il più noto cugino del C/64

di **Domenico Pavone**

La tecnica del Raster è indispensabile per visualizzare qualunque cosa su qualunque computer

Cominciamo subito col precisare che, in tema di Raster, quanto è stato più volte descritto sulla rivista (soprattutto dal n. 47 in poi) a proposito del C/64, in linea di massima mantiene intatta tutta la sua validità anche se applicato al C/128.

A parte le considerazioni teoriche, ovviamente uguali, risultano infatti identiche anche le principali locazioni da manipolare per smanettare con il Raster, ovvero quelle inerenti il famigerato VIC (Video Interface Controller).

Se comunque appartenete alla sparuta fazione... integralista (128 o morte!) che ignora qualunque articolo ad esso non strettamente dedicato, niente paura: si comincia subito con una rispolverata generale alle cognizioni di base, adattandole al nostro caro e bistrattato C/128 (in modo 128).

VIC, INTERRUZIONI ED ALTRO

Il VIC, com'è noto, non è altro che un circuito adibito ad una serie di funzioni riguardanti il video, come la visualizzazione ed il controllo degli sprite, l'alta risoluzione, e varie altre cosette tutte riservate allo schermo su 40 colonne (per quello a 80 colonne provvede il VDC).

In pratica, si tratta di una serie di registri (= locazioni specializzate) allocati in Bank 15 da $D000 (53248) a $D030 (53296), per un totale di 49.

E qui appare la prima differenza con il VIC del C/64, formato da "sole" 47 locazioni, le stesse prime 47 del C/128 (poste, inoltre, agli stessi indirizzi!).

Il sovrannumero, nel 128, dipende dalla necessità di gestire la scansione di un maggior numero di tasti (registro 48), nonchè dalla possibilità di operare in modo "fast" (registro 49).

Per il resto, come già detto, i vari registri operano in modo analogo al C/64, anche se, nel loro utilizzo, occorre tenere presente alcune particolari procedure interne, diverse nei due computer.

Vedremo comunque, tra breve, come le diversità si risolvano spesso in una maggiore facilità di programmazione quando si opera in modo 128 (almeno in questo caso).

Un'altra indispensabile premessa, prima di parlare più direttamente del Raster e dei suoi effetti, riguarda gli interrupt.

Anche questo è un tema che dovrebbe essere più che noto ai lettori di questa rivista, ma alcune peculiarità del 128, legate proprio al rapporto tra le interruzioni ed il nostro ancora misterioso Raster, sono forse meno conosciute.

Ma procediamo con ordine.

Gli Interrupt del microprocessore 8502 (C/128), così come quelli del 6510 (C/64), vengono in genere divisi in due categorie: quelli "mascherabili" (IRQ) e quelli "non mascherabili" (NMI).

In pratica, significa che i primi possono essere ignorati dal sistema (è proprio quello che provochiamo con il comando Assembly SEI), mentre gli altri hanno una priorità assoluta.

Quando interviene una richiesta di Interrupt, come sappiamo, il computer "molla" quello che sta facendo per svolgere una serie di compiti che dipendono dal tipo di interruzione occorsa.

Poco da dire sulle NMI, se non che sono collegate alla pressione del tasto Restore (ed eventualmente a Run / Stop), o alla ricezione di un segnale dalla porta seriale RS-232.

Molto più attinenti al nostro argomento sono, invece, gli IRQ.

E' proprio grazie agli IRQ "di sistema" che il computer svolge la maggior parte dei suoi compiti, che

Mentre Primo Giovedini fa notare al comandante che due guerrieri barbari su una portaerei c'entrano più o meno come il SID c'entra con le schermate Koala, un aereo sta volando: è quello del tenente JACK ASSEMBLER...

```
                  DISASSEMBLATO 1
=============================================

01           *=$1300      ;Start a 4864.
02 ;-----------------------------------------
03           SEI          ;Disabilita interr.
04           LDA #<ROUT   ;Inserisce indir.
05           LDX #>ROUT   ;della routine
06           STA $314     ;nel vettore di
07           STX $315     ;IRQ (low/hi).
08           CLI          ;Abilita interr.
09           RTS          ;Return.
10 ;-----------------------------------------
11 ROUT      LDA $D019    ;Controlla bit 0
12           AND #$01     ;di VICIRQ.
13           BEQ NMIEXT   ;Se bit0 = 0, esce.
14           STA $D019    ;Se bit0 = 1, cont.
15           LDA $FB      ;Prende colore e lo
16           STA $D021    ;"poka" nello sfondo.
17           INC $FB      ;Colore = colore + 1.
18           LDA $FC      ;Flag $FC in A.
19           CLC          ;Si aggiunge 8 al
20           ADC #$08     ;flag $FC e alla
21           STA $FC      ;linea di scansione
22           STA $D012    ;del Raster reg.
23           BCC NMIEXT   ;Se linea<255, esce.
24 ;-----------------------------------------
25 IRQEXT JMP $FA65       ;Ritorno a IRQ.
26 NMIEXT JMP $FF33       ;Uscita comune da IRQ
```

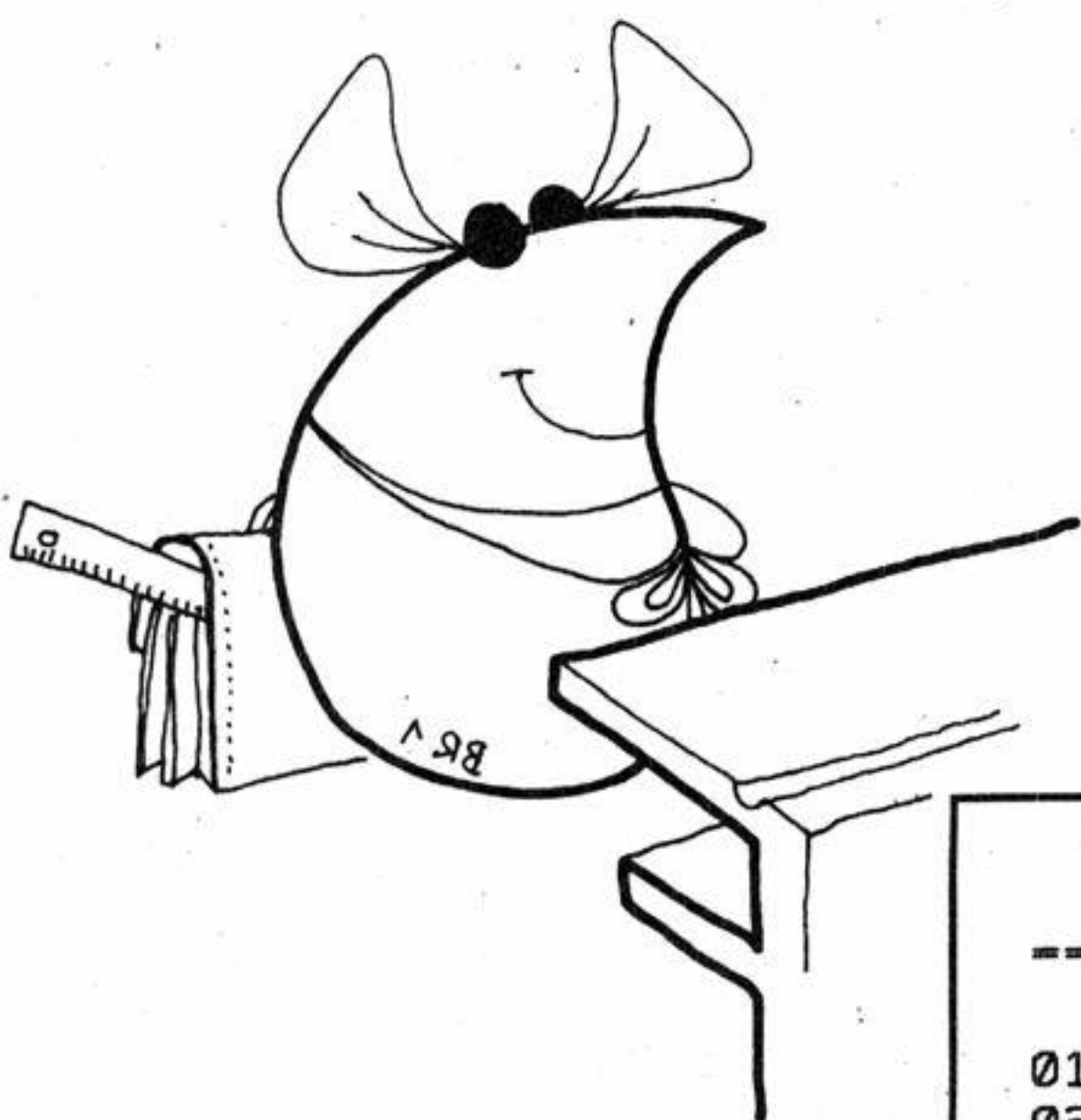
nel C/128 sono decisamente numerosi: oltre alla scansione della tastiera, al controllo del cursore e varie altre mansioni già note ai sessataquattristi, in modo 128 il sistema deve anche controllare quale porzione di schermo gestire in modo testo (ci si può trovare in "Split screen" per effetto dei comandi Graphic 2 oppure 4), e vari altri parametri legati a comandi Basic, tipo Collision, Movspr, Sound, Play, eccetera.

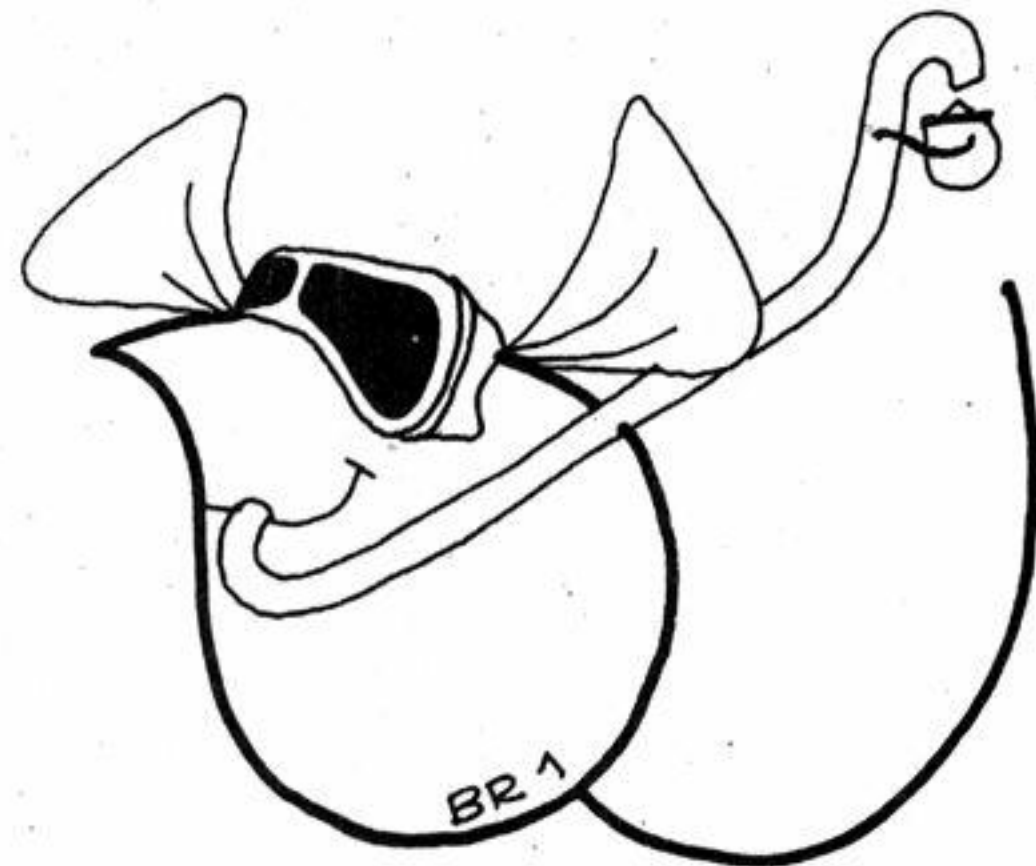
Tutti questi compiti, è più che risaputo, vengono svolti ogni cinquantesimo di secondo, allorchè, appunto, perviene al sistema una richiesta "interna" di Interrupt.

Già, ma che si intende per "interna"?

Nel buon vecchio C/64, l'IRQ veniva attivato da una struttura hardware, uno dei timer del CIA; nel 128, invece, il sistema implementato è diverso: l'Interrupt, o meglio la richiesta di Interrupt, scatta ogni qualvolta il pennello (o cannone) elettronico dello schermo scandisce una determinata linea.

Ed eccoci giunti al nocciolo della questione.

Le differenze tra il C/64 ed il C/128 sono minime

DISASSEMBLATO 2

```
====================================================
01          *=$1300     ;Start a 4864.
02  ;----------------------------------------------
03          SEI         ;Disabilita interr.
04          LDA #<ROUT  ;Inserisce indir.
05          LDX #>ROUT  ;della routine
06          STA $314    ;nel vettore di
07          STX $315    ;IRQ (low/hi).
08          CLI         ;Abilita interr.
09          RTS         ;Return.
10  ;----------------------------------------------
11  ROUT    LDA $D019   ;Controlla bit 0
12          AND #$01    ;di VICIRQ.
13          BEQ NMIEXT  ;Se = 0, esce...
14          STA $D019   ;Se = 1, continua.
15          LDA $FB     ;Prende colore e lo
16          STA $D020   ;"poka" nel bordo.
17          INC $FB     ;Colore=colore + 1.
18          LDA $FC     ;Flag $FC in A.
19          CLC         ;Si aggiunge 8 al
20          ADC #$08    ;flag $FC e alla
21          STA $FC     ;linea di scansione
22          STA $D012   ;del Raster reg.
23  ;----------------------------------------------
24          LDX $FD     ;X = linea raster.
25          BEQ CONT    ;Salta se linea<255.
```

```
26          CMP #$38   ;Linea = 57?
27          BCC NMIEXT ;Se minore, esce.
28 ;-----------------------------------
29          LDA $08    ;Setta al valore 8
30          STA $FB    ;il colore e il
31          STA $FC    ;flag di posizione.
32          DEC $FD    ;Linea raster < 255.
33          LDA $D011  ;Azzera il bit 7
34          AND #$7F   ;del registro 53265
35          STA $D011  ;(bit 9 del Raster).
36          LDA #$00   ;Azzera contenuto
37          STA $D012  ;del Raster reg.

38 ;-----------------------------------
39 NMIEXT JMP $FF33   ;Uscita da IRQ.
40 ;-----------------------------------

41 CONT    BCC NMIEXT ;Se linea <255,esce.
42 ;-----------------------------------

43          LDX #$00   ;Azzera  il flag
44          STX $FC    ;di posizione.
45          INC $FD    ;Linea raster > 255.
46          LDA $D011  ;Setta ad 1 il nono
47          ORA #$80   ;bit del Raster
48          STA $D011  ;register e...
49          STX $D012  ;azzera Raster.
50 ;-----------------------------------
51 IRQEXT JMP $FA65   ;Uscita IRQ standard.
```

Una volta recuperato Jack Assembler sulla portaerei si discute dell'accaduto.
Negli ultimi tempi si erano fatti vivi alcuni "pirati" di software, amici del famoso G.B. MOX, che alla fine avevano iniziato a dare un po' troppo fastidio a Giovedini e amici...

Non è possibile manipolare il Raster in Basic, ma occorre operare con il l.m.

SUA MAESTA' IL RASTER

Intanto... gonfiamo il petto di sacro orgoglio mediterraneo: caso più unico che raro, ci troviamo di fronte non all'ennesimo termine inglese, ma ad un vocabolo latino.

"Raster", infatti, sta per "rastrello", con un preciso riferimento alle tracce parallele lasciate sul terreno da questo strumento.

Passando dal giardinaggio all'elettronica, per Raster si intende l'insieme di linee parallele che compongono un'immagine su un qualsiasi tipo di apparecchio video.

Una schermata, infatti, è data dalla rapidissima scansione orizzontale di un certo numero di linee (313 per il sistema europeo), effettuata da uno strumento chiamato, come già detto, cannone elettronico.

Ultimata l'immagine, il cannone punta di nuovo in alto a sinistra del video e ripete la scansione.

Tutto ciò, manipolando opportunamente alcuni registri del VIC e redirigendo il normale percorso dell'Interrupt di sistema, consente realizzazioni altrimenti impossibili per il computer.

Entriamo ora nei dettagli e, soprattutto, "amalgamiamo" la serie di informazioni finora riferite in ordine sparso.

Il VIC mantiene una informazione costante sulle linee di scansione del video attraverso uno dei suoi registri, locato (sempre nella Ram di Banco 15) in $D012 (53266), e chiamato (guarda caso) Raster Register.

Tentando la lettura del valore contenuto in tale locazione, l'operazione restituisce il numero della linea raster scandita in quel momento; se, al contrario, si tenta di scrivere un valore, tale operazione provocherà l'immagazzinamento, dello stesso valore, in un altro speciale registro interno, detto di comparazione.

Poichè, inoltre, una locazione non può assumere valori al di sopra di 255 (un byte rimane sempre un byte!), per le linee da 256 a 312 risulterà settato anche un altro bit (praticamente il nono del Raster register), il più alto del registro $D011: in concreto, per valori superiori a 255, il bit 7 di $D011 sarà settato, mentre $D012 conterrà il numero della riga diminuito di 256.

Gli effetti di un'operazione in scrittura, possono dunque essere così riassunti:

- Ogni volta che la linea di scansione del Raster è uguale al valore inserito tramite il registro $D012 (più il bit 7 di $D011), viene settato ad 1 il bit 0 del registro $D019 (53273 = Interrupt register), il cui compito è, appunto, quello di precisare qual'è stata la fonte di un interrupt.

- Se il bit meno significativo del registro $D01A (53274 = Abilitazione degli interrupt) risulta settato, viene effettivamente provocato un interrupt quando la scansione del raster raggiunge la linea in questione.

Complicato?

Beh, non è proprio semplicissimo, ma certamente più a parole che nei fatti.

UN VIDEO PSICHEDELICO

Cominciamo col vedere in azione una prima e semplice applicazione di quanto si è finora detto. Si copi il listato 1; poi, dopo averlo salvato su periferica, si impartisca Run.

Dopo una frazione di secondo, necessaria per allocare in memoria il programma (necessariamente) in linguaggio macchina, riapparirà il Ready su uno sfondo decisamente allegro.

Provando (per esempio) a dare List, si può notare come lo sfondo rimanga inalterato, mentre il normale funzionamento del computer non viene minimamente compromesso.

Per ripristinare la situazione di default, basterà premere Run / Stop + Restore.

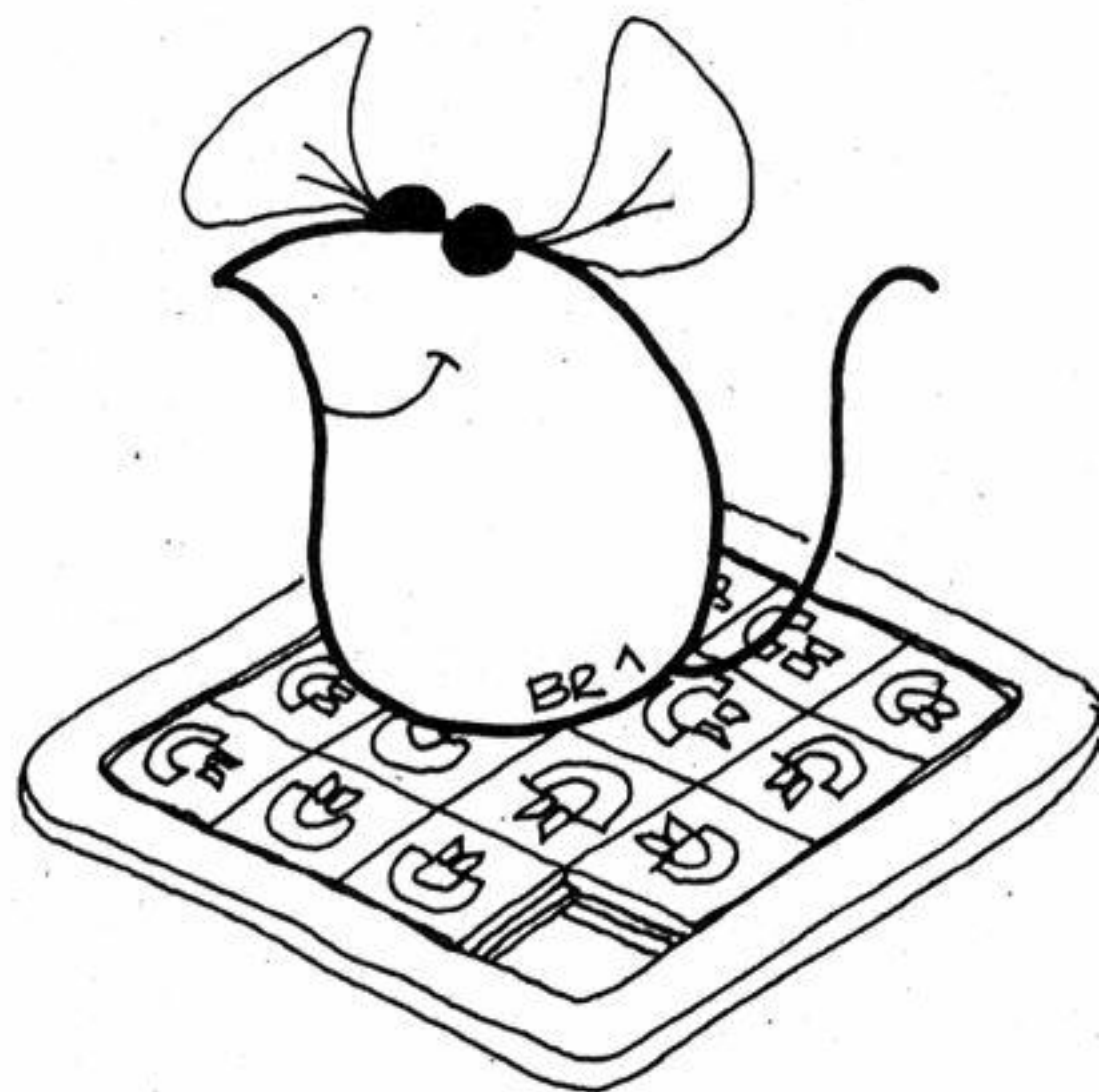
Il tutto, tra l'altro, grazie ad una routine (disassemblato 1), molto più breve di quanto sarebbe stato necessario per ottenere lo stesso scopo sul C/64.

Analizzandola più a fondo se ne comprenderà il perchè.

Il primo gruppo di istruzioni (righe 3 - 9), si limita a dirottare il vettore di Interrupt all'inizio della routine vera e propria (riga 11).

E' questa una tecnica che dovrebbe essere più che nota, per cui limitiamoci a precisare che, nel C/128, l'indirizzo cui il sistema deve tornare per il normale interrupt è $FA65 (64101), mentre è possibile uscire dal ciclo IRQ anche tramite un salto all'indirizzo $FF33 (65331).

In quest'ultimo caso vengono scavalcate le normali procedure di IRQ, e semplicemente ripristinati il contenuto originario dei registri, nonchè la configurazione di banco.

```
10 REM -----------------------------
15 REM    RASTER C/128:   LISTATO 1
20 REM -----------------------------
25 :
30 COLOR 4,1: COLOR 5,2:BANK 15
35 FAST:FOR X=0 TO 47:READ A
40 B=B+A: POKE 4864+X, A:NEXT
45 IF B<>5625 THEN SLOW: GOTO 55
50 SLOW: SYS 4864: END
55 PRINT"ERRORE!":SLOW:END
60 :
65 DATA 120,169,013,162,019,141,020
70 DATA 003,142,021,003,088,096,173
75 DATA 025,208,041,001,240,025,141
80 DATA 025,208,165,251,141,033,208
85 DATA 230,251,165,252,024,105,008
90 DATA 133,252,141,018,208,144,003
95 DATA 076,101,250,076,051,255
96 END
```

Unica precauzione da rispettare quando si gioca con l'Interrupt, è che il nostro programma "cuneo" sia presente (e quindi visibile) nella Ram di banco 15, nel nostro caso quella libera da interferenze, posta a partire da $1300 (4864).

Vediamo dunque ciò che succede quando, ogni cinquantesimo di secondo, viene eseguita la nostra routine.

Anzitutto viene testato (tramite AND, riga 12) il bit 0 del registro $D019, per appurare se si è giunti alla linea di scansione raster inserita in $D012.

In caso affermativo (bit = 1), si preleva il colore da $FB e lo si inserisce nella locazione $D021 (53281 = colore sfondo).

Poi, dopo avere aggiornato il colore incrementando $FB, con le linee 18 - 22 si somma 8 al numero di riga raster memorizzato tramite $D012.

Quindi, se la somma ha dato come risultato un valore inferiore a 255, si esce con un salto a $FF33, altrimenti il JMP sarà diretto verso il normale IRQ.

In pratica, pur non essendo qui presente alcun riferimento alle già accennate locazioni $D01A e $D011, si provoca ugualmente un interrupt ogni volta che il raster giunge alla riga da noi voluta (quella "storata" in $D012).

Il motivo è presto detto: non dovendo superare la linea 255, viene sfruttato il fatto che il sistema provvede da solo ad azzerare il bit 7 di $D011 (il "nono" del Raster register).

Inoltre, e questa è forse la più grossa differenza col C/64, non occorre avvisare (settando il bit 0 di $D01A) per produrre un interrupt alla linea da noi memorizzata nel Raster register: dato che anche il sistema adopera il raster per l'interrupt principale, provvederà "lui" (o meglio, il Kernal) a inizializzare (ad 1) il bit in questione.

NON SOLO SFONDO

Per quanto simpatico, uno sfondo variopinto non è proprio il massimo della comodità, almeno se poi si vuole adoperare normalmente lo schermo.

Proviamo dunque a trasferire la stessa procedura sul bordo.

Sono a disposizione numerosi registri per gestire correttamente il Raster

```
10 REM -------------------------------------------------------
15 REM  R A S T E R   C / 128:   L I S T A T O  2
20 REM -------------------------------------------------------
25 FAST:BANK 15:COLOR0,1:COLOR5,16:FORX=0TO96:READ A:B=B+A
30 POKE4864+X,A:NEXT:IFB<>12868THENPRINT"ERRORE!":SLOW:END
35 KEY1,"POKE4890,32"+CHR$(13):KEY3,"POKE4890,33"+CHR$(13)
40 SLOW:SYS4864:END
45 DATA 120,169,013,162,019,141,020,003,142,021,003,088,096,173
50 DATA 025,208,041,001,240,049,141,025,208,165,251,141,032,208
55 DATA 230,251,165,252,024,105,008,133,252,141,018,208,166,253
60 DATA 240,028,201,056,144,021,165,008,133,251,133,252,198,253
65 DATA 173,017,208,041,127,141,017,208,169,000,141,018,208,076
70 DATA 051,255,144,251,162,000,134,252,230,253,234,234,234,173
75 DATA 017,208,009,128,141,017,208,142,018,208,076,101,250
80 END
```

Adoperando la stessa routine, basterà modificare l'istruzione di riga 16 (sempre del disassemblato 1) in STA $D020, direttamente col monitor, o più semplicemente con una Poke 4890, 32.

Le righe colorate si trasferiranno sì sul bordo... ma fino ad un certo punto.

La parte inferiore dello schermo resterà, per così dire, scoperta, com'era logico aspettarsi.

Infatti, con la nostra routine, non si aveva bisogno di gestire l'ormai noto nono bit del Raster register, che ora invece diventa necessario.

Per vederlo in azione, si resetti il computer, quindi si copi il listato 2 e lo si mandi in esecuzione.

Se tutto è in regola (= se non avete commeso errori), ora il bordo dello schermo sarà uniformemente riempito dalle strisce colorate.

Premendo inoltre F1 ed F3, si potrà passare dalla colorazione del bordo a quella dello sfondo (a voi il compito di movimentare ulteriormente le cose).

Ad un primo esame, il disassemblato (2) della nuova routine appare più complesso, e questo perchè gli stessi aspetti che prima semplificavano la manipolazione del Raster, ora richiedono una certa attenzione.

Vediamoli in sintesi.

La prima parte della routine (righe 1 - 22) è praticamente identica alla precedente, e non necessita di altri chiarimenti.

Alle righe 24 - 27, si è aggiunto il controllo della locazione $FD, che servirà da indicatore: se posta ad 1, significherà che il Raster ha superato la linea 255.

In questo caso occorrerà che, ad ogni passaggio dell'interrupt, venga settato il bit più alto di $D011 (righe 46 - 48), che poi, dopo la posizione 312, dovrà essere riazzerato (righe 33 - 35).

In entrambi i casi andrà azzerato anche il contenuto del registro $D012 (36 - 37 e 49), mentre le locazioni $FB, $FC ed $FD andranno resettate di conseguenza.

Le operazioni non sono comunque così difficili, e possono agilmente essere esaminate con l'aiuto dei commenti al disassemblato.

Per concludere, non resta che ricordare come, una volta appresi questi primi rudimenti sul Raster del C/128, diventano possibili una infinità di sofisticate applicazioni: non ultime, quelle già proposte dalla rivista a prosito del C/64 (16 sprite, sprite sul bordo, ecc.).

Le differenze, ormai l'avrete capito, non sono poi così abissali, e, soprattutto, giocano in favore di voi (noi) 128isti.

QUANDO UNA PAGINA NON BASTA

Un nostro lettore ha sofisticato un listato, pubblicato in precedenza, adattandolo anche al C/128; vediamo di che si tratta

di **Alfonso Daniele**

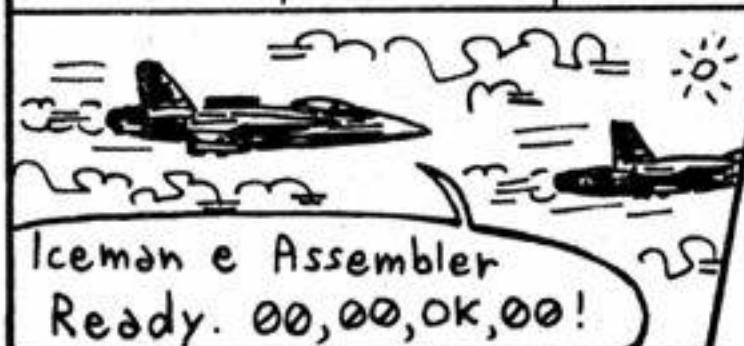
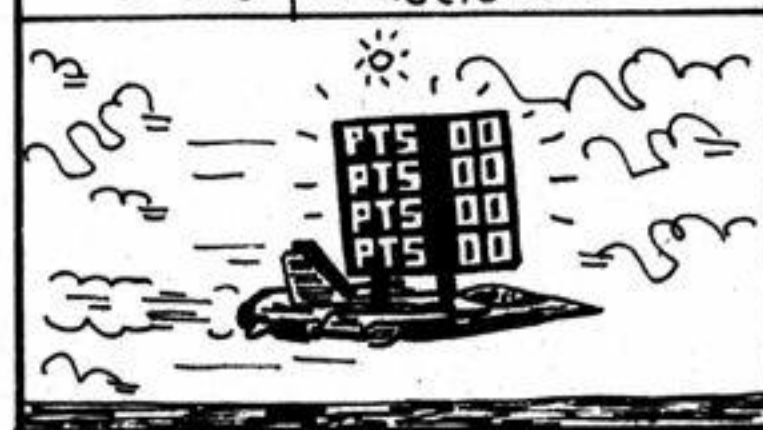
Il programma di queste pagine rappresenta un miglioramento rispetto a quello proposto da Domenico Pavone sul numero 54 di C.C.C. ("Uno schermo di scorta").

Le modifiche introdotte sono due: la prima consiste nell'utilizzo del tasto funzione F1 per l'attivazione del programma (che evita, quindi, di "sporcare" lo schermo con eventuali SYS); la seconda permette lo scambio di DUE pagine video in bassa risoluzione.

Per mezzo del precedente programma, infatti, era possibile memorizzare una sola schermata (mediante un'opportuna SYS) e richiamarla, poi, in modo del tutto simile. Al momento del richiamo della pagina precedentemente memorizzata, però, quella in uso veniva irrimediabilmente persa.

Studiando i listati altrui è facile raggiungere buoni risultati

```
10 REM *** PROGRAMMA UNO ***
20 REM SWITCH SCREEN C-64
30 REM BY DANIELE ALFONSO
40 :
50 FOR X=49152 TO 49224:READA:C=C+A:POKEX,A:NEXT
60 IF C<>11629 THEN PRINT"ERRORE NEI DATA.":END
70 SYS49152:PRINT"SWITCH-SCREEN ATTIVATO.":END
80 DATA 169,11,141,20,3,169,192,141
90 DATA 21,3,96,165,203,201,4,208
100 DATA 3,32,23,192,76,49,234,169
110 DATA 0,170,133,249,133,251,133,253
120 DATA 169,201,133,250,169,4,133,252
130 DATA 169,196,133,254,160,0,177,251
140 DATA 145,249,177,253,145,251,177,249
150 DATA 145,253,200,208,241,232,230,250
160 DATA 230,252,230,254,224,4,208,228
170 DATA 96
180 END
```

```
10 REM *** PROGRAMMA QUATTRO ***
20 REM DEMO PER SWITCH-SCREEN C-64
30 REM BY DANIELE ALFONSO
40 :
50 PRINT"{CLR}":FOR X=1 TO 12
60 PRINT"{DOWN}SALVE A TUTTI I LETTORI DI C.C.C."
70 NEXT
80 SYS 49175
90 PRINT"{CLR}":FOR X=1 TO 12
100 PRINT"{DOWN}{RVS}SALVE A TUTTI I LETTORI DI C.C.C."
110 NEXT
120 SYS 49175
130 FOR X=1 TO 200:NEXT
140 GOTO 120
```

Per utilizzare i tasti funzione è necessario interagire con l'Interrupt

I listati di queste pagine, pertanto, risolvono il problema grazie allo scambio simultaneo di due diverse pagine testo, in modo che nessuna delle due vada persa durante lo scambio.

COME FUNZIONA

Per illustrare l'algoritmo usato ricorreremo ad una esemplificazione che chiama in gioco due soli byte, chiamati BYTE1 e BYTE2, di cui vogliamo scambiare i valori.

Leggendo il valore di BYTE1, e memorizzandolo subito in BYTE2, verrebbe perso il valore che quest'ultimo possiede al momento dello scambio. Abbiamo quindi bisogno di un terzo byte, che funga da buffer, in cui depositare temporaneamente uno dei due valori. Il nuovo byte in questione sarà chiamato BYTE3.

A questo punto, per scambiare tra loro il contenuto di BYTE1 e BYTE2, non dovremo far altro che:

- leggere il contenuto di BYTE1 e salvarlo momentaneamente in BYTE3;
- leggere il valore di BYTE2 e depositarlo in BYTE1;
- leggere il valore di BYTE3 (corrispondente al precedente valore di BYTE1) e depositarlo in BYTE2.

Tutto chiaro?

L'algoritmo appena descritto è precisamente quello utilizzato nel programma l.m. di cui potete esaminare sia il listato "caricatore" in Basic sia il disassemblato (formato Macroassembler Commodore): l'area video viene memorizzata nel buffer; subito dopo la pagina video "nascosta" viene trasferita nella memoria video; infine il contenuto del buffer passa nella zona nascosta.

La procedura viene eseguita a gruppi di 256 byte per 4 volte di seguito (256 x 4 = 1 Kbyte), usando l'indirizzamento indicizzato in pagina zero, in cui sono stati precedentemente depositati i puntatori al buffer ($F9 - $FA), quelli alla memoria video ($FB - $FC) e quelli alla zona nascosta ($FD - $FE).

All'inizio del programma vengono modificati i puntatori degli interrupt per farli puntare alla routine di controllo dell'ultimo tasto premuto. Il controllo viene eseguito mediante la lettura della locazione $CB che contiene, appunto, tale valore. Se questo è uguale a 4 allora vuol dire che è stato premuto il tasto F1 e, di conseguenza, la routine di Switch-screen viene attivata; in caso contrario si salta alla normale routine di gestione degli interrupt ($EA31).

Si noti come la sezione del programma che si occupa dello switch sia stata definita come una subroutine e termina, pertanto, con un comando RTS.

```
10 REM *** PROGRAMMA TRE ***
20 REM SWITCH-SCREEN VERSIONE C-128
30 REM BY DANIELE ALFONSO
40 :
50 FOR X=3072 TO 3144:READA:POKE X,A:C=C+A:NEXT
60 IF C<>11032 THEN SCNCLR:PRINT"ERRORE NEI DATA.":END
70 SCNCLR:KEY1,"":PRINT"SWITCH-SCREEN ATTIVATO.":SYS3072:END
80 DATA 169,11,141,20,3,169,12,141,21
90 DATA 3,96,165,212,201,4,208,3,32,23
100 DATA 12,76,101,250,169,0,170,133
110 DATA 250,133,252,133,254,169,4,133
120 DATA 251,169,32,133,253,169,36,133
130 DATA 255,160,0,177,250,145,252,177
140 DATA 254,145,250,177,252,145,254
150 DATA 200,208,241,232,230,251,230
160 DATA 253,230,255,224,4,208,228,96
170 END
```

Questo particolare permette di richiamarla non solo in modo diretto, tramite la pressione di F1, ma anche in modo programma, mediante SYS 49175.

Per essere usato in modo diretto, mediante gli interrupt, il programma richiede la consueta SYS 49152, essendo allocato nell'usatissima area a disposizione del programmatore l.m. Poichè tale zona è di ben 4 Kbyte, anche il buffer e la zona nascosta sono allocati in quest'area, ed esattamente dalle locazioni $C900 e $C400 in poi.

Per chi possiede un C/128 viene riportata anche la versione di Switch-screen per questo computer, di cui potete trovare il caricatore Basic nel programma tre.

Le modifiche apportate, che non condizionano assolutamente il funzionamento del listato, sono le seguenti:

- il programma è allocato dalla locazione $0C00 in poi, owerossia nell'area dedicata al buffer di input della RS-232.
- il salto alla normale gestione degli interrupt è verso $FA65.
- il buffer e l'area nascosta sono stati allocati, per motivi di semplicità, nell'area dedicata alla pagina grafica in hi-res; un eventuale comando GRAPHIC 1,1 (pur non essendo pericoloso) è quindi da evitarsi, pena la cancellazione dei dati memorizzati.

La routine pubblicata, a dispetto della sua brevità, svolge una funzione molto utile

```
10 REM *** PROGRAMMA CINQUE ***
20 REM DEMO SWITCH-SCREEN C-128
30 REM BY DANIELE ALFONSO
40 :
50 SCNCLR:DIRECTORY:SYS3095:SCNCLR:LIST
60 DO:SYS 3095:SLEEP 1:LOOP
```

```
1000 ; ** PROGRAMMA DUE **
1010 ; SWITCH-SCREEN IN MACROASSEMBLER
1020 ; BY DANIELE ALFONSO
1030 ;
1040 *=$C000  ; ALLOCATO DA 49152
1050 LDA #$0B ; DIROTTA
1060 STA $314 ; I PUNTATORI
1070 LDA #$C0 ; DEGLI
1080 STA $315 ; INTERRUPT
1090 RTS      ; TORNA AL BASIC
1100 LDA $CB  ; ULTIMO TASTO PREMUTO
1110 CMP #$04 ; E' = 4?
1120 BNE SALTO; SE NO: VAI A SALTO
1130 JSR SUB  ; SE SI': ESEGUI IL PROGRAMMA
1140 SALTO JMP $EA31 ; SALTA ALLA NORMALE GESTIONE DEGLI INTERRUPT
1150 SUB LDA #$00 ; AZZERA ACCUMULATORE
1160 TAX ; AZZERA REGISTRO X
1170 STA $F9 ; LO-BYTE BUFFER =00
1180 STA $FB ; LO-BYTE AREA VIDEO =00
1190 STA $FD ; LO-BYTE ZONA 2 =00
1200 LDA #$C9; HI-BYTE BUFFER =C9
1210 STA $FA
1220 LDA #$04; HI-BYTE AREA VIDEO =04
1230 STA $FC
1240 LDA #$C4; HI-BYTE ZONA 2 =C4
1250 STA $FE
1260 CICLO LDY #$00 ; CICLO TRASFERIMENTO
1270 START LDA ($FB),Y ; METTE VIDEO
1280 STA ($F9),Y ; NEL BUFFER
1290 LDA ($FD),Y ; METTE ZONA 2
1300 STA ($FB),Y ; NEL VIDEO
1310 LDA ($F9),Y ; METTE BUFFER
1320 STA ($FD),Y ; NELLA ZONA 2
1330 INY ; Y = Y + 1
1340 BNE START ; Y=0? SE NO VAI A START
1350 INX ; X = X + 1
1360 INC $FA ; HI-BYTE BUFFER + 1
1370 INC $FC ; HI-BYTE AREA VIDEO + 1
1380 INC $FE ; HI-BYTE ZONA 2 + 1
1390 CPX #$04 ; X = 4 ?
1400 BNE CICLO ; SE NO: RIPETI CICLO
1410 RTS ; SE SI': ESCI DAL PROGRAMMA
```

I lettori più bravi non troveranno comunque difficoltà nello spostare buffer e area due nella zona, assai più tranquilla, posta da $12FE in poi.

L'argomento della SYS, per utilizzare Switchscreen 128 in modo programma, è 3095.

Troverete due banali esempi di applicazione (programmi demo "quattro" e "cinque", rispettivamente per C/64 e C/128) ma che rappresentano solo l'inizio: starà a voi trovare l'uso migliore dei listati pubblicati.

Noi, a tal proposito, abbiamo tante di quelle idee...

SCONVOLGIMENTI INFORMATICI

Una tecnica di codificazione dei nostri file più segreti, possibile solo grazie alla versatilità dell'Assembly

di **Domenico Pavone**

Cominciamo subito col rassicurare il lettore abituale di questa rivista (gli altri non si porranno certo il problema): se teme di trovarsi di fronte all'ennesimo sistema di protezione di un programma, ebbene... ha quasi indovinato.

La tecnica che esamineremo tra breve, infatti, consente sì una protezione "mirata" dei programmi ma, per le sue caratteristiche, può anche applicarsi ad altri tipi di file, come i testi elaborati da un word processor.

In altre parole, più che un sistema di protezione, vedremo in azione una tecnica di codificazione, ovvero qualcosa di simile alla cifratura di un messaggio (vedi n.52 pagina XIII di Campus).

Il tutto, com'è prassi, fungerà da movente (niente omicidi, però) per approfondire le istruzioni di scorrimento e rotazione in Assembly, per cominciare ad usare le cosiddette "Macro" del Macro Assembler, nonchè per una bella indigestione di Kernal.

Ce n'è abbastanza per venire subito al sodo.

UN FILE DA ROVISTARE

Qualunque file (e quindi un programma, un testo o un archivio) non è altro che una sequela di numeri, che il computer interpreta differentemente a seconda dell'ambito in cui essi vengono gestiti.

Se, per esempio, si considera un progamma Basic, questo è composto da una serie di byte (non dimentichiamo che il computer "ragiona" solo in binario), disposti in un certo ordine, che l'interprete del Basic gestirà poi come link, numeri di linea, token, e così via.

Analogamente, anche un file di testo è formato da soli numeri: toccherà poi ad un word processor interpretarli come codici Ascii e trattarli di conseguenza.

Se, dunque, prendiamo un file e ne modifichiamo i byte che lo compongono, questo risulterà incomprensibile per il software che lo gestisce, sia esso l'interprete del Basic, un word processor, o altro.

E' chiaro, però, che il file risulterà anche inutilizzabile.

Ma se le modifiche apportate sono reversibili, ecco che si dispone di una vera e propria chiave di accesso.

Un esempio banale: se ad ogni byte che compone un programma (o un testo) si somma 10, questo risulterà inservibile per tutti, tranne per chi è al corrente che prima di attivarlo è necessario sottrarre 10 ad ogni byte del file.

Attenzione, però: un byte, è arcinoto, può assumere solo valori compresi tra 0 e 255, quindi l'operazione di somma e sottrazione appena citata risulta piuttosto scomoda da attuare in pratica. Consideriamo, ad esempio, il caso della codifica di un byte, di valore 250, contenuto in una cella di memoria. Se tentassimo (in Basic) di "pokare" da qualche parte il risultato della sua somma con 10, otterremmo solo un inevitabile Illegal Quantity Error.

Una prima, "classica" soluzione, tipicamente Assembly, consiste nello sfruttare l' operatore logico EOR, con le stesse modalità della routine proposta sul n.57 per porre in reverse lo schermo.

Effettuando l'Or Esclusivo di ogni byte di un file con un valore, predeterminato, che funge da "chiave" (compreso tra 0 e 255), e ripetendo la stessa operazione con lo stesso valore-chiave, il file verrà decodificato.

Questo grazie alla proprietà di EOR di restituire lo stesso valore originario, se si effettua due volte l'operazione con lo stesso operando.

Una volta deciso di ricorrere all'Assembly, comunque, anche l'operazione di somma prima ac-

Per codificare un programma è utile sfruttare particolari istruzioni del L.M.

LOAD E SAVE A TUTTO KERNAL

Le operazioni di caricamento e salvataggio su disco (o nastro), estremente semplici e lineari quando si opera in Basic, in Assembly sembrano assumere una difficoltà che tuttavia è solo apparente.

In fondo anche in Basic, per ottenere un minimo di versatilità da un Load o Save, è spesso necessario ricorrere a trucchi non sempre facili da gestire, come la modifica dei puntatori da 43 a 46.

Al contrario, adoperando correttamente le routine del Kernal preposte all'esecuzione di Load e Save, si può fare di tutto, e senza eccessivo sforzo.

Passsando alla pratica, vediamo come vanno usate.

Entrambe, prima di essere richiamate, richiedono che vengano precisati alcuni parametri tramite due routine del Kernal:
SETLFS ($FFBA)
SETNAM ($FFBD)

Con SETLFS vanno settati i parametri riguardanti il file, preparando i registri del 6510 come segue...

Accumulatore = Numero logico del file (compreso tra 0 e 255).
Registro X = Numero della periferica (8 per il drive, 1 per il tape).
Registro Y = Indirizzo secondario (Ne riparleremo tra poco).

...e poi saltando con un JSR all'indirizzo $FFBA.

Con SETNAM, invece, vengono segnalati al sistema i dati riguardanti, appunto, in nome del file. Si avrà in questo caso:
Accumulatore = Lunghezza del nome del file.
Registro X = Byte basso dell'indirizzo dove è depositato il nome del file.
Registro Y = Byte alto del suddetto indirizzo.
Anche qui, sarà poi necessario un JSR $FFBD.

A questo punto, tutto è pronto per il Load o Save veri e propri.

Per il caricamento, si procede come segue:
• Caricare l'Accumulatore con il valore 0 (con 1 si effettua un'operazione Verify).
• Se con SETLFS si è impostato a 0 il registro Y, allora il file (indipendentemente dalla sua testata, contenente l'indirizzo di caricamento) verrà caricato a partire dall'indirizzo specificato in X e Y, che andranno quindi preparati con ripettivamente il byte basso e quello alto della locazione desiderata.
• Se, invece, con SETLFS si è usato un indirizzo secondario 1, il file verrà caricato a partire dalla locazione indicata dal contenuto dei suoi primi due byte (l'header, o testata che dir si voglia): in questo caso il contenuto di X e Y è irrilevante.
• Richiamare la routine Load con JSR $FFD5, tenendo presente che, dopo il caricamento, si avrà in X e Y l'indirizzo dell'ultima locazione occupata dal file.

Per effettuare un Save, ecco invece i passi necessari:
• Preparare due locazioni in pagina zero con l'indirizzo d'inizio dell'area di memoria da salvare, e porre in Accumulatore il numero della prima delle due locazioni.
• Caricare in X (byte basso) e Y (byte alto) l'indirizzo dell'ultima locazione dell'area di memoria da salvare.
• Saltare alla routine di Save con JSR $FFD8.

MACRO COME... MACROASSEMBLER

Una delle caratteristiche che fanno dell'Editor / Assembler Commodore uno strumento davvero professionale, è senza dubbio la possibilità di inserire, negli elaborati, le cosiddette Macro, o, più chiaramente, Macro Istruzioni.

Si tratta, per l'appunto, di una serie di comandi Assembly, racchiusi tra due particolari pseudo-istruzioni:

.MAC
.MND

La prima indica l'inizio della Macro, mentre la seconda la conclude.

L'uso di questa feature è più semplice di quanto si creda, e risulterà ancora più comprensibile se andiamo a dare un'occhiata al disassemblato proposto in queste pagine.

Forse avrete già notato come, nelle righe 69 e 77, sia presente un'istruzione decisamente al di fuori del comune: Incr.

In effetti non si tratta di un comando dell'Assembly, ma semplicemente del nome della Macro presente in riga 13.

Quindi, regola prima: Una Macro viene attivata semplicemente inserendo il suo nome (di vostra scelta) nella posizione in cui normalmente si inserisce uno mnemonico dell'Assembly (Incr, nel *nostro* caso).

Il nome viene assegnato digitandolo subito dopo l'istruzione .MAC (riga 13).

Giunta in riga 69, quindi, la routine LM passa ad eseguire la Macro (come pure dopo riga 77) ma, a differenza di una normale subroutine, agirà nei due casi in maniera differente.

Dopo Incr, infatti, sono specificate due label:
CD, SALV (in riga 69)
DCD, LNK (in riga 77).

Con tale procedura si "passano" alla Macro dei parametri, di volta in volta diversi, che vengono accettati all'interno della Macro stessa inserendo, al posto dell'operando di un'istruzione, un punto interrogativo seguito da un numero (righe 15, 19, 20): con 1 verrà inserita la prima label specificata dopo Incr, con 2 la seconda, e così via (per un massimo di ben 9 parametri!).

Nel nostro disassemblato, la Macro consente di non riscrivere due volte la stessa routine che incrementa Y ed i due puntatori di pagina 0: dopo Incr (di riga 69) il salto alle righe 15 e 20 avverrà all'indirizzo precisato da CD (quindi alla riga 64) mentre in riga 19, in pratica, è come se fosse presente un comando BEQ SALV.

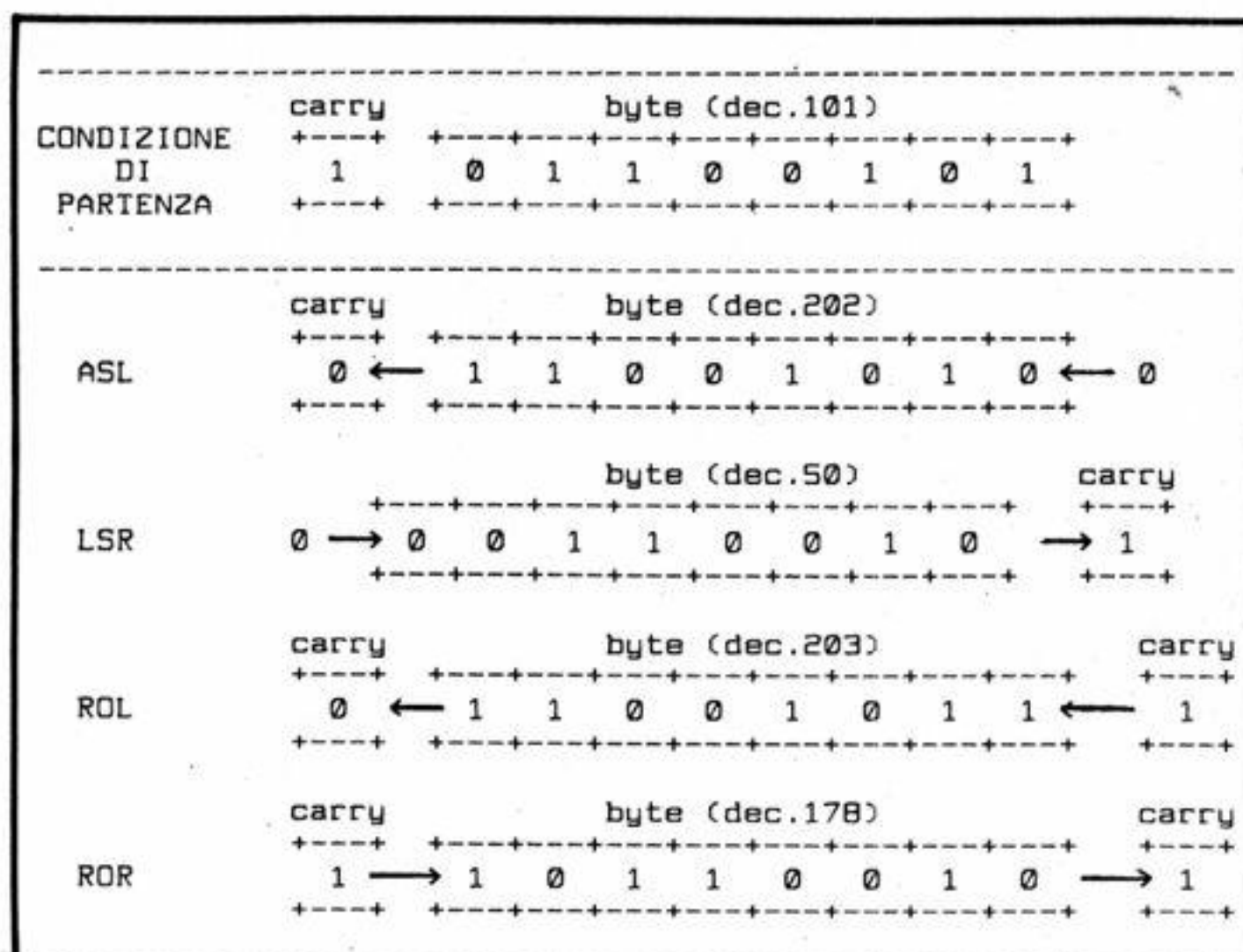
Dopo l'INCR di riga 77, gli stessi salti saranno invece diretti verso le due label DCD e LNK.

Si badi, però, che nel codice macchina finale, cioè dopo l'assemblaggio, il gruppo di istruzioni poste all'interno della Macro sarà effettivamente presente due volte.

L'Assembler, cioè, ogni volta che incontra un richiamo alla Macro, la inserisce (tutta) in quel *punto*, pur se con i parametri da noi specificati.

In effetti, dunque, non si tratta di una vera e propria subroutine... ma al programmatore va più che bene lo stesso.

PROCURIAMOCI IL MACRO ASSEMBLER COMMODORE

Per procurarsi il package "Macro Assembler Commodore" è sufficiente recarsi presso un qualsiasi negozio specializzato (soprattutto se Commodore Point) oppure rivolgersi presso le Ditte che pubblicano pagine di pubblicità sulla nostra stessa rivista.

L'Or esclusivo è alla base di quasi tutti i processi di codificazione

cennata diverrebbe realizzabile, e le tecniche possibili diventano davvero numerose.

Vediamone in particolare una, che può in ogni caso fungere da base di lancio per eventuali sviluppi più sofisticati e, soprattutto, maggiormente personalizzati.

CHIUSURA... LAMPO

Cominciamo col copiare il listato 1, quindi salviamolo su disco (non si tralasci quest'ultima operazione!).

Come si può notare, la routine è interamente in linguaggio macchina: il listato è solo un caricatore che, dopo il Run, alloca la routine in memoria a partire dalla locazione 49152.

Chi dispone del MacroAssembler, può evitare la noiosa sequela di Data e "farsi la mano" copiando, con l'editor, il disassemblato 1 (i commenti si possono tralasciare), che sarà poi necessario assemblare e caricare in memoria.

Dopo aver lanciato il programma Basic, la routine l.m. è pronta per essere attivata secondo le due modalità illustrate sullo schermo: con...
Sys 49152, 1
...avverrà la codifica del programma Basic (o l.m, purchè residente in area Basic) al momento in memoria, mentre con...
Sys 49152, 0
...si attiverà la decodifica.

In pratica, per cifrare o decifrare un programma occorrerà prima caricarlo (load...,8), poi impartire una Sys 49152 seguita dalla virgola e dal valore (0 o 1) che si desidera.

Supponendo di avere ancora in memoria il listato caricatore (assicuratevi di averne salvata una copia: in caso di errore si rischia la sua perdita), digitiamo Sys 49152, 1 e Return.

Un messaggio chiederà il nome con il quale archiviare su disco il progranna codificato. Se non si desidera salvarlo, basta premere Return.

Nel nostro caso, limitiamoci a premere Return.

Quasi immediatamente, non si dimentichi che ci si trova di fronte a Sua Maestà LM, riapparirà il Ready.

Se ora si prova a chiedere il List, otterremo, sullo schermo, solo dei caratteri senza senso, o addirittura il computer sembrerà inchiodarsi.

Premiamo dunque Run / Stop + Restore, e impartiamo Sys 49152, 0.

Scegliamo ancora l'opzione "No Save" premendo Return, e riproviamo poi con List: riecco il nostro listato originale in tutto il suo splendore.

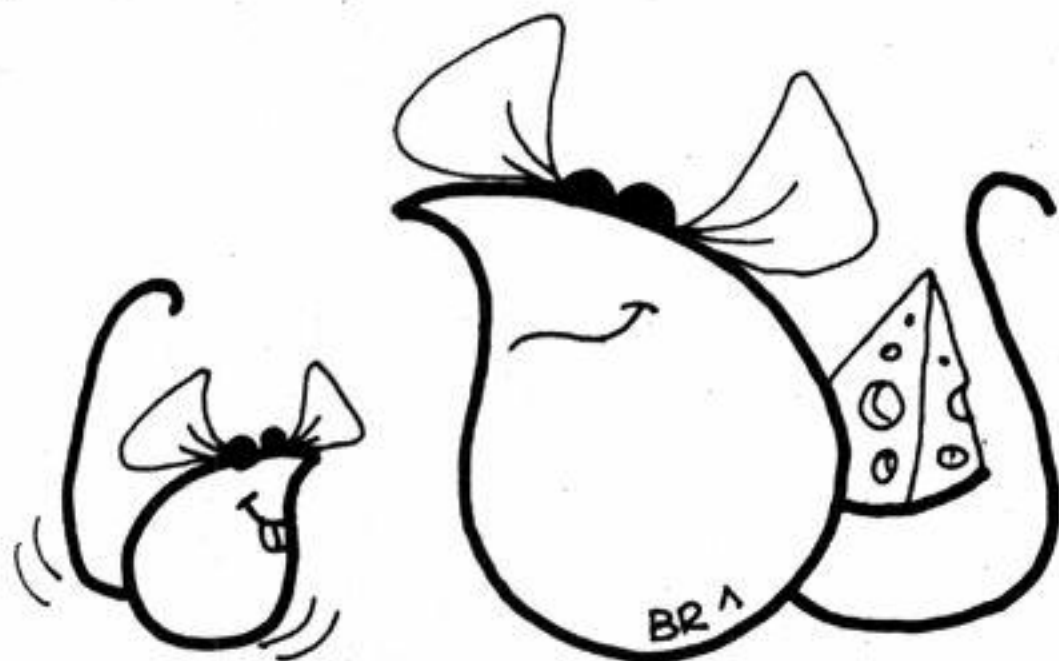
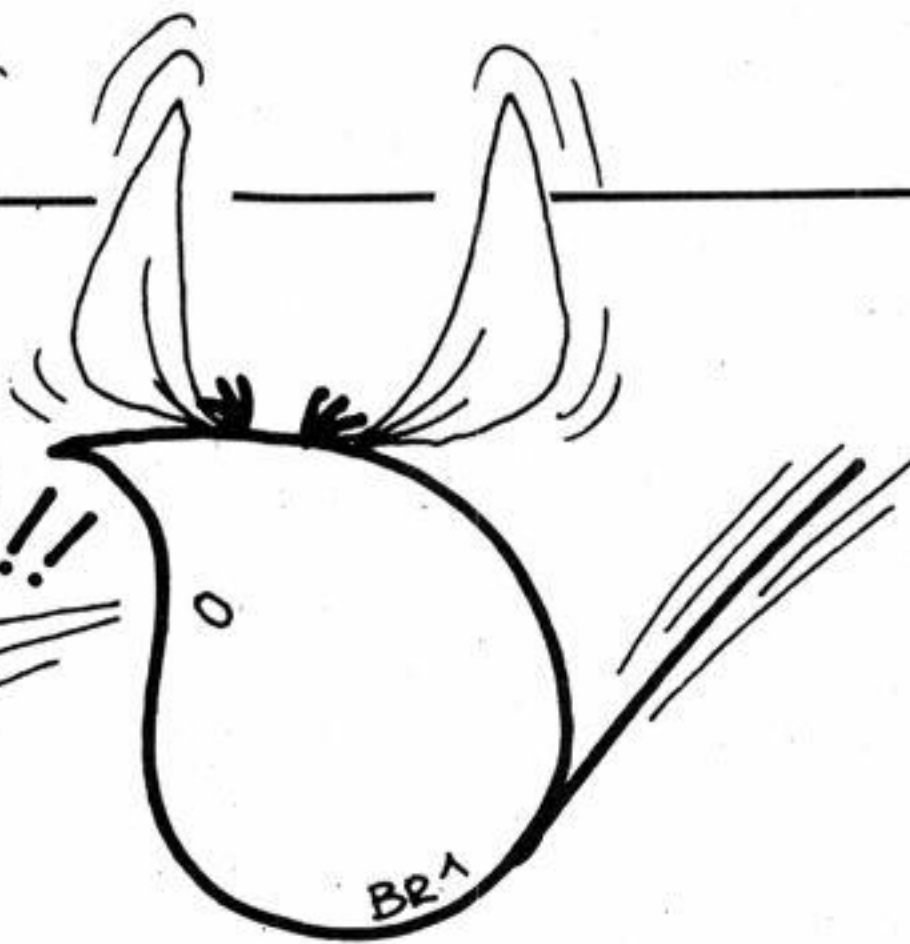
Com'è ovvio, se si vuole riservare l'utilizzo di uno o più programmi (parleremo tra breve dei testi) solo a determinate persone, basterà fornirle della routine di codifica / decodifica e delle versioni "cifrate" salvate su disco.

Pur non utilizzando vere e proprie "password" (= parole chiave), la procedura di codifica / decodifica può essere resa ancora più ermetica applicandola più di una volta, ma questo risulterà chiaro dopo un'attento esame dello sviluppo Assembly della routine.

I PRELIMINARI

Facendo riferimento al disassemblato 1 (in formato Macro Assembler), il programma vero e proprio inizia dalla riga 23. La sezione che occupa le linee 13 - 21, successive alle varie inizializzazioni, rappresentano una Macro istruzione, analizzata nel riquadro specifico di queste pagine.

Tornando al primo gruppo di comandi (righe 23 -

```
01 PUNT    =$FB            ;dec.251
02 ENDPRG  =$2D            ;dec.45
03 INPUT   =$ABFD          ;dec.44029
04 READY   =$A474          ;dec.42100
05 RELINK  =$A533          ;dec.42291
06 CHROUT  =$FFD2          ;dec.65490
07 SETLFS  =$FFBA          ;dec.65466
08 SETNAM  =$FFBD          ;dec.65469
09 SAVE    =$FFD8          ;dec.65496
10 BUFFER  =$200           ;dec.512
11         *=$C000         ;start a 49152
12 ;====================================
13         .MAC INCR       ;Inizio Macro.
14         INY             ;Y=Y+1
15         BNE ?1          ;Se y<>0 salta.
16         INC PUNT+1      ;Incrementa byte
17         LDA PUNT+1      ;alto puntatote
18         CMP $FA         ;e lo compara.
19         BEQ ?2          ;Se uguale, esce.
20         BNE ?1          ;Se no, continua.
21         .MND            ;Fine Macro.
22 ;====================================
23         JSR $AEFD       ;Legge virgola e
24         JSR $B79E       ;valore dopo SYS.
25         CPX #$2         ;Se >= a 2,
26         BCS USCITA      ;esce, altrimenti
27         STX $02         ;deposita valore.
28 ;------------------------------------
29         LDX #$00        ;Legge caratteri
30 PRINT   LDA MESG,X      ;per l'input fino
31         BEQ NOMPRG      ;allo zero e li
32         JSR CHROUT      ;stampa sul video.
33         INX             ;Bne qui funziona
34         BNE PRINT       ;come un Jmp.
35 ;------------------------------------
36 NOMPRG  JSR INPUT       ;Routine Input.
37         LDX #00         ;Legge nome file
38 LENFIL  LDA BUFFER,X    ;da buffer basic.
39         BEQ CONT        ;0 =fine del nome.
40         INX             ;In X len(nome).
41         BNE LENFIL      ;Salta (Jmp).
42 ;------------------------------------
43 CONT    TXA             ;A=len nome file.
44         LDX #<BUFFER    ;In X e Y indiriz
45         LDY #>BUFFER    ;zo nome file.
46         JSR SETNAM      ;Routine Kernal.
47 ;------------------------------------
48         LDX #$8         ;Num. periferica.
49         LDY #$FF        ;Nessun comando.
50         JSR SETLFS      ;Routine Kernal.
51 ;------------------------------------
52         LDA #$01        ;Byte basso del
53         STA PUNT        ;puntatore.
54         LDX #$8         ;Byte alto del
55         STX PUNT+1      ;puntatore.
56         LDX ENDPRG+1    ;Byte alto fine
57         INX             ;programma + 1
58         STX $FA         ;in locazione 250.
59         LDY #$00        ;Y = 0.
60 ;------------------------------------
61         LDA $02         ;Biforcazione per
62         BEQ DCD         ;codifica o decod.
63 ;------------------------------------
64 CD      CLC             ;Carry = 0.
65         LDA (PUNT),Y    ;Prende byte,
66         ROL A           ;lo ruota a sin.,
67         ADC #$00        ;somma il carry e
68         STA (PUNT),Y    ;lo rideposita.
69         INCR CD,SALV    ;Aggiunge Macro.
70 ;------------------------------------
71 DCD     CLC             ;Carry = 0.
72         LDA (PUNT),Y    ;Prende byte,
73         ROR A           ;lo ruota a des.
74         BCC STORE       ;Se carry=0,salta.
75         ORA #$80        ;Setta bit 7.
76 STORE   STA (PUNT),Y    ;Rideposita byte.
77         INCR DCD,LNK    ;Aggiunge Macro.
78 ;------------------------------------
79 LNK     JSR RELINK      ;Link linee basic.
80 SALV    LDA #$2B        ;A=punt. inizio.
81         LDX ENDPRG      ;LowByte fine prg.
82         LDY ENDPRG+1    ;HiByte fine prg.
83         JSR SAVE        ;Routine Kernal.
84 ;------------------------------------
85 USCITA  JMP READY       ;Torna al basic.
86 ;------------------------------------
87 MESG    .BYT 'NOME FILE DA SALVARE '
88         .BYT '(RETURN = NO SAVE)'
89         .BYT 141,0
90         ..END
```

Il set di istruzooni del 6510 prevede anche la rotazione dei bit in un byte

27), vediamo qui applicata una tecnica presa in prestito dalle routine dell' Enciclopedia L.M.

Si tratta, in pratica, del salto all'indirizzo di due utilissime routine dell'interprete del Basic: la prima ($AEFD) semplicemente controlla che il carattere cui punta il sistema sia una virgola (nel nostro caso, la virgola che segue Sys 49152). Se non la trova, si avrà un immediato ritorno in ambiente Basic con un bel Syntax Error.

La seconda routine di sistema ($B79E), controlla invece che, dopo la virgola, sia presente un valore numerico, compreso tra 0 e 255, che deposita nel registro X.

Facile, dunque, fare in modo che il valore accettato sia solo 0 oppure 1 (righe 25 - 26); questo viene depositato nella locazione 2 di pagina zero per poi essere sfruttato, in riga 61 - 62, come indicazione di codifica (cd) o decodifica (dcd).

Sbrigata questa prima formalità, con le righe da 29 a 41 la nostra superveloce routine realizza un vero e proprio Input, sfruttando la sezione dell'interprete Basic preposta a tale comando ($ABFD).

Questa opera semplicemente stampando un punto interrogativo (se non lo si vuole, basta usare $AC00 come ingresso); in seguito, con la pressione del Return, immette la linea digitata nel buffer del Basic, allocato a partire da $200 in poi.

Da tener presente che sarà uno zero a segnalare la fine della stringa di input.

Poichè l'Assembly non dispone dei lussi di alto livello (come nel Basic) prima di utilizzare Input è necessario stampare il messaggio relativo (righe 29 - 34), che nel disassemblato è letto dalle righe 87 - 89 e visualizzato sullo schermo tramite la routine del Kernal Chrout (tecnica già usata sul n.58).

Il codice Ascii 141 (riga 89) provvede solo a simulare la pressione di Shift + Return, per mandare a capo il cursore.

A questo punto disponiamo, nelle locazioni da $200 in poi, del nome del file da salvare (se si è risposto all'input), ma, ai fini di un successivo Save, sarà necessario segnalare al sistema anche la lunghezza del nome del file (vedi riquadro specifico). Per ricavarla (l'Assembly non dispone di un comando Len come il Basic), ecco entrare in azione le istruzioni poste alle righe 38 - 41, molto semplici da comprendere col solo aiuto dei commenti nel disassemblato.

Ed ecco che, settati i vari parametri del file (righe 43 - 50 e riquadro), si entra nel cuore della routine.

```
01  ;=================
02     DISASSEMBLATO 2
03  ;=================
04           *=49152
05  ;--- CODIFICA ----
06           CLC
07           LDA $FB
08           ROL A
09           ADC #$00
10           STA $FB
11           RTS
12  ;-- DECODIFICA ---
13           CLC
14           LDA $FB
15           ROR A
16           BCC STORE
17           ORA #$80
18  STORE    STA $FB
19           RTS
20           .END
```

GIROTONDO DI BIT

Anzitutto, con le linee da 52 a 58, viene delimitato il campo di azione.

In pratica si preparano due puntatori di pagina zero ($FB - $FC) con l'indirizzo di inizio del programma da codificare (nel formato byte basso / byte alto), mentre per l'indirizzo di fine programma viene usato solo il byte alto, non essendo necessaria una determinazione esatta anche nel byte basso.

```
100 REM ---------------------------------------------------
110 REM      CODIFICA / DECODIFICA PROGRAMMI IN AREA BASIC
120 REM ---------------------------------------------------
130 FOR X=0 TO 171: READA:POKE49152+X,A:B=B+A:NEXT:PRINTCHR$(147)
140 IFB <> 21911 THEN PRINT "ERRORE NELLE LINEE DATA!!": END
150 PRINT"SYS 49152,1 = CODIFICA"
160 PRINT"SYS 49152,0 = DECODIFICA"
170 END
180 DATA 032,253,174,032,158,183,224,002,176,118,134,002,162,000
190 DATA 189,131,192,240,006,032,210,255,232,208,245,032,253,171
200 DATA 162,000,189,000,002,240,003,232,208,248,138,162,000,160
210 DATA 002,032,189,255,162,008,160,255,032,186,255,169,001,133
220 DATA 251,166,046,232,134,250,162,008,134,252,160,000,165,002
230 DATA 240,021,024,177,251,042,105,000,145,251,200,208,245,230
240 DATA 252,165,252,197,250,240,028,208,235,024,177,251,106,144
250 DATA 002,009,128,145,251,200,208,243,230,252,165,252,197,250
260 DATA 240,002,208,233,032,051,165,169,043,166,045,164,046,032
270 DATA 216,255,076,116,164,078,079,077,069,032,070,073,076,069
280 DATA 032,068,065,032,083,065,076,086,065,082,069,032,040,082
290 DATA 069,084,085,082,078,032,061,032,078,079,032,083,065,086
300 DATA 069,041,141,000
310 END
```

Già qui, volendo, è possibile apportare eventuali modifiche: invece dell'indirizzo $0801 (= inizio del Basic), se avete voglia di smanettare col Macro Assembler, si può inserire nei puntatori l'inizio di programmi LM, magari prelevandolo (dopo un Open) dai primi due byte dei file registrati su disco.

Anche l'indirizzo di fine (nella routine: $2D = fine programma Basic) può essere modificato prelevandolo dalle locazioni $AE - $AF (decimale 174 - 175), che indicano l'ultima locazione caricata dopo un Load.

Ma torniamo al nostro disassemblato.

In riga 64, inizia la codifica. Viene prelevato il byte da trattare, quindi, prima di essere "ripokato" nella stessa locazione tramite un indirizzamento indiretto (vedi C.C.C. n. 58), viene sottoposto ad una rotazione a sinistra dall'istruzione ROL (ROtate Left)

Vediamo di chiarire che cosa avviene, rifacendoci anche alla figura 1.

Con le istruzioni di scorrimento (ASL e LSR) e di rotazione (ROL e ROR) è possibile spostare di una posizione verso destra o sinistra tutti i bit di un byte.

Con gli scorrimenti viene inserito uno zero nel bit rimasto vuoto, mentre il bit uscente va a finire nel Flag di riporto (Carry).

Adoperando invece le istruzioni di Rotazione, nel bit vuoto viene inserito il contenuto del carry che, dopo la rotazione, conterrà il bit uscente.

Nella nostra applicazione, dopo una rotazione a sinistra (riga 66), è necessario non perdere il bit uscente (che si trova nel carry), altrimenti risulterebbe poi impossibile una decodifica del file.

Ecco allora che, dopo ROL, viene sommato l'eventuale Carry (ADC) al byte in questione, cosicchè

Codificare un file di testo è semplice come la codifica di un programma

```
10 REM -------------------------------
12 REM   CODIFICA / DECODIFICA TESTI
14 REM -------------------------------
16 PRINTCHR$(147)
18 FORX=0TO19:READA:POKE49152+X,A:NEXT
20 INPUT"CODIFICA O DECODIFICA (C/D)";A$
22 IFA$<>"C"ANDA$<>"D"THEN20
24 INPUT"NOME DEL FILE DA TRATTARE";F$
26 INPUT"TIPO DI FILE (S,P,U)";T$
28 IFT$<>"S"ANDT$<>"P"ANDT$<>"U"THEN26
30 INPUT"NOME DEL FILE TRATTATO";SF$
32 REM -------------------------------
34 OPEN15,8,15:OPEN5,8,5,F$+","+T$+",R"
36 INPUT#15,E,E$:IFE<>0THENPRINTE$:GOTO52
38 OPEN6,8,6,SF$+","+T$+",W"
40 INPUT#15,E,E$:IFE<>0THENPRINTE$:GOTO52
42 GET#5,B$:IFST=64THEN52
44 B=ASC(B$+CHR$(0)):POKE251,B
46 IFA$="C"THENSYS49152:REM--- CODIFICA
48 IFA$="D"THENSYS49161:REM- DECODIFICA
50 PRINT#6,CHR$(PEEK(251));:GOTO42
52 CLOSE5:CLOSE6:CLOSE15:END
54 REM ---------- ROUTINE LM ----------
56 DATA 024,165,251,042,105,000,133,251
58 DATA 096,024,165,251,106,144,002,009
60 DATA 128,133,251,096
70 END
```

Con il sistema descritto è possibile effettuare sette codifiche diverse per ciascun file.

quello che era il bit più significativo, dopo un girotondo passando per il Carry, diventerà il bit 0, ovvero il meno significativo.

Per la decodifica (righe 71 - 77), basterà dunque effettuare l'operazione inversa: ruotare a destra il bit, e, in base al contenuto del Carry, settare o meno l'MSB (Most Significative Bit = bit 7) con l'istruzione ORA di riga 75.

L'operazione di codifica o decodifica viene poi ripetuta per tutti i byte del programma, incrementando l'indice Y e i puntatori tramite la Macro di nome Incr specificata alle righe 13 - 21 (vedi riquadro), quindi si passa al salvataggio del file tramite le istruzioni alle linee 80 - 83.

Infine, come d'obbligo, un ritorno al Basic, ma tramite un salto alla già vista (n.58) routine $A474.

Il nostro C/64, infatti, non ha certo la memoria corta, e non dimentica che dopo la Sys di attivazione abbiamo aggiunto di soppiatto una virgola e un numero: con un normale "rientro" (RTS), ce lo ricorderebbe stampando uno sgradevole Sintax Error.

UN'ALTRA CHANCE

A questo punto i più attenti avranno già mormorato: "Già, ma non si doveva parlare anche di codifica di testi?"

Una momenta, bitte!

Il sistema di codifica / decodifica fin qui visto è applicabile, senza alcuna reale modifica, ad un file di testo.

L'unica differenza, ovviamente, consiste nell'accesso al file.

In altre parole, se per trattare un programma lo abbiamo prima caricato in memoria e poi salvato, per un file di testo (Seq, Prg o User che sia) occorrerà prima aprirlo in lettura, prelevarne i singoli byte, "ruotarli", ed infine riscriverli su un diverso file, precedentemente aperto in scrittura.

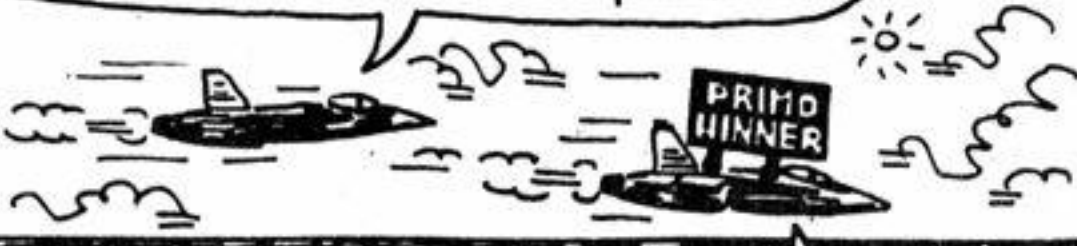
C'è però da considerare che, anche ricorrendo all'Assembly, le operazioni di accesso alla periferica sono decisamente lente, per cui risulta forse più semplice sfruttare il buon vecchio Basic, con una subroutine LM che compia solo il lavoro di codifica / decodifica.

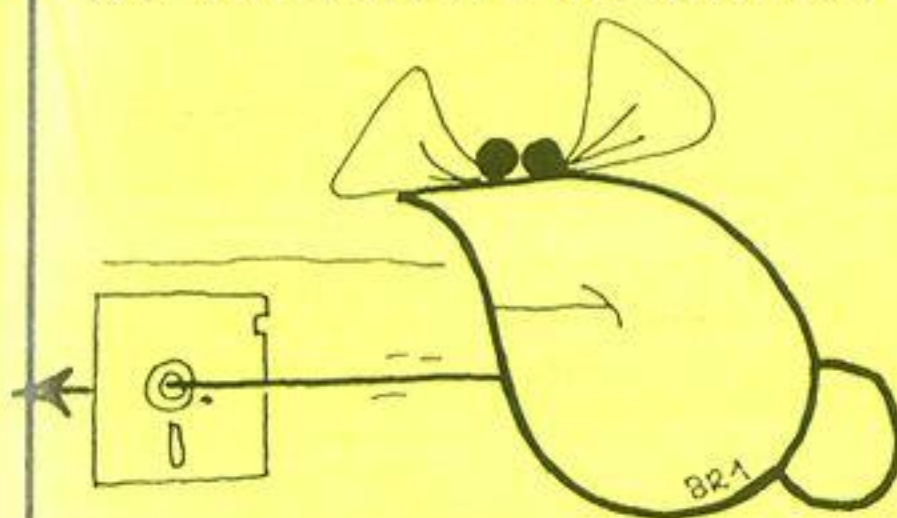
Un esempio di come procedere è proposto dal listato 2 (e relativo disassemblato) che dovreste essere ormai in grado di comprendere anche senza commenti.

In effetti, con il programma di listato 2, è possibile trattare anche programmi Basic o LM, ma, come potrete constatare sperimentandolo, con attese decisamente scomode se paragonate all'efficacia della precedente routine.

Si ricordi che è possibile effettuare più volte (dopo 7 però si torna al punto di partenza) la codifica dello stesso file, cui dovrà ovviamente corrispondere un egual numero di decodificazioni.

Un ultimo consiglio prima di concludere:

Se intendete sfruttare questo sistema per inviare dischetti "roventi" ad eventuali partner... clandestini, assicuratevi prima che, dalle sue parti, non circolino altri lettori di Commodore Computer Club!

```
10 REM FINE DEL FUMETTO
20 PRINT"VUOI RILEGGERE LA STORIA?"
30 GETA$:IFA$<>"Y"ANDA$<>"N"THEN30
40 IFA$="Y"THEN GOTO (INIZIO INSERTO)
50 IFA$="N"THEN SYS 64738

READY.
█
```

VUOI PUBBLICARE UN ANNUNCIO GRATUITO?

Commodore Computer Club ti offre la possibilità di pubblicare GRATUITAMENTE, e immediatamente dopo che lo spedisci, il tuo annuncio riguardo scambio o vendita di software, vendita o acquisto di apparecchi usati, ricerca di amici per fondare un club e così via. Per fare in modo che il tuo annuncio compaia nel prossimo numero della rivista siamo costretti a seguire una procedura rapidissima e dobbiamo quindi chiederti una piccola cortesia: compila le righe che seguono usando un grosso pennarello (o "ripassandole" più volte con la penna), tenendo presente che IL TUO STESSO ANNUNCIO verrà ridotto, per mezzo di procedimenti fotografici, a cm. 7 x 3 circa e pubblicato così come ci perviene in redazione, senza alcuna modifica. Ti consigliamo, pertanto, di usare una calligrafia chiara, a stampatello e dotata di caratteri sufficientemente grandi per consentire un'agevole lettura anche dopo il "trattamento" di riduzione.

Invia l'annuncio in busta chiusa, affrancata secondo le norme vigenti, indirizzando a:

Commodore Computer Club
Viale Famagosta, 75
20142 MILANO

CERCO I SEGUENTI PROGR. PER C 64
COUNTING BEE, A BEE C'S, THE SPELLING
BEE, THE MAGIC GARDEN TALKING BOOK,
VOCABOLARIO PER CARTUCCIA MAGIC VOICE
COMBA PIER LUIGI 011 3081388

SCAMBIO SOFTWARE DI OGNI
TIPO PER C64. CERCO PROGRAMMI
PER STAMPARE E DI MUSICA.
MARCO RAUSA - VIA CA' BIANCA
3/7 - 40131 BOLOGNA

CERCO UN RAGAZZO\A POSSESORE
DI C=64 X SCAMBIARE
SOFTWARE. TELEFONATE A
743460 0332
CHIEDETE DI REMOLO
Ciao

VENDO MANNESMANN TALLY 80 + PERFETTA A MIGLIOR
OFFERENTE - PER AMIGA 500 NOVITÀ SETT. A PREZZI RIDICOLI
VENDO 400 DISK PIENI A/B PER C-64 COMPRESE NOVITÀ A
LIRE 1500 SUPPORTO COMPRESO - MAX SERIETÀ
ANFOSSO MAURIZIO - VIA SIMONETTI 4 - 14018 ROATTO - TEL. 0141/938118

COMPRO - DISK DRIVE 1541
(VITALI ANDREA - VIA CARLO
BERTINAZZI 25 - 00139 ROMA
TEL. 06/8138526).

CERCO: OUT RUN - IMPOSSIBLE
MISSION 1-2-3 - SPY VS SPY -
SU CASSETTA, PER C.64.
SCRIVERE A: LAZZARINO OSCAR
VIA. VITTONE 18, CHIERI (TO)

SOLO AMICI 64 GRATIS ESAME TEMA
NATALE - INDICATE DATA, LUOGO, ORA DI
NASCITA E COMPUTER CLUB - DR. SEB.
CATALANO - METELLO 3 - AGRIGENTO

X C.64 - 3000 TITOLI SCAMBIO - VENDO
X VIDEOLISTA + PRG - SPEDIRE N° 1
DISCHETTO + PRG INSERITI + BOLLO
SAMANNÀ GIOVANNI V. MAZZINI N. 24
91027 PACECO (TP) TEL. 0923 - 882848

VENDO, CAMBIO, COMPRO SOFTWARE PER AMIGA!
SCAMBIO ESPERIENZE DI PROGRAMMAZIONE!
MASSIMA SERIETÀ! RISPOSTA ASSICURATA!!!
ANDREA FRADEANI, CORSO CRIMEA N.35
ALESSANDRIA 15100 TEL. 0131/63864

VENDO UN ADATTATORE
TELEMATICO, MODELLO
6499 PER CBM 64 A £77000
TELEFONARE A RICCARDO
CASTELLANI 0586/852048

MI CHIAMO CHRISTIAN ACERBI
ABITO IN VIA CASORETTO 44 (MILANO)
E VORREI VENDERE UN DATASSETTE PER PLUS/4 1531 SEMI NUOVO

DESIDERO METTERMI IN CONTATTO CON APPASSIONATI
DI COMPUTER C64 PER SCAMBIO IDEE, SOFTWARE
E NOTIZIE SU DISCO 1541 O CASSETTA.
BOCCACCI FABIO · VIA OGNIBENE 4B - 43100 PARMA
GRAZIE PER LA COLLABORAZIONE!

VENDO PROGRAMMI SCOLASTICI SU FLOPPY
PER C/64 PER RAGAZZI 6-11 ANNI
CHIEDERE ELENCO E CARATTERISTICHE
AGAZZI CESARE VIA BRAVI 18/A
24030 MAPELLO (BG)

VENDO HARDWARE-SOFTWARE PER
C-64/128 A PREZZO IRRISORIO.
VENDO STAMPANTE MPS 1200.
TELEFONARE IN ORE PASTI
AL 0862/66468 - MAURIZIO

FINALMENTE ANCHE SUL TUO C128
VERI PROGRAMMI IN CP/M+ E
GWBASIC C128 E C64
TELEFONA 0883/24738 DALLE
ORE 14 ALLE 16 E ORE SERALI

VENDO C64, DRIVE 1541C+TURBO SPEEDDOS
REGISTRATORE, JOYSTICK, TASTIERINO MUSICALE
2 CORSI DI BASIC, OLTRE 20 RIVISTE, CIRCA
400 PROGRAMMI (UTILITY E VIDEOGAMES).
L. 650.000 TEL. 02/4582278

CERCHIAMO APPASSIONATI POSSESSORI DI C64 PER FONDARE UN NUOVO
CLUB. TUTTI I SOCI RICEVERANNO UNA "RIVISTA" MENSILE CON TUTTE
LE NOVITA' PER IL LORO COMPUTER. INOLTRE OGNI NUOVO ISCRITTO
POTRA' AVERE A DISPOSIZIONE 4 VIDEOGAMES A SUA SCELTA IN OMAGGIO!
PER INFORMAZIONI CONTATTATECI SUBITO! Pierangelo Galizia
via Appia S.S. 7 n. 3 - 85050 Baragiano Sc. (PZ) -
Tel. 0971/993258 (h. 16:00 - 19:00)

CERCO SOFTWARE (SU CASS.)
MUSICALE PER C-64
GIOVANNI VICIDOMINI
VIA GRAMSCI 7
84015 NOCERA SUP.

CAMBIO PROGRAMMI PER "COMMODORE-64"
SOLO SU DISCO
MASSIMA SERIETÀ E VELOCITÀ
SCRIVERE A: CANE CIRO - VIA ARCORA C.MLE
N: 7 - 80013 CASALNUOVO (NA) - 081/8421438

SONO DISPOSTO A SCAMBIARE SOFT.
E IDEE X CBM64. MASSIMA SERIETÀ
GIANPAOLO RUOTOLO, P.ZZA DE GASPERI
N° 13, 71100 FOGGIA. TEL 0881-37388

VENDIAMO GIOCHI E PRG.
DI OGNI TIPO (SOLO FLOPPY)
PER C 64/128. SOLO FIRENZE.
£ 3'500 PER OGNI TITOLO. TEL. PASTI:
494783 OMAR o 487823 GIANLUCA

PER C=64 VENDO E SCAMBIO
PROGRAMMI, GIOCHI E UTILITY SU
DISCO E CASSETTA
TEL. ORE PASTI: 28-80-07 (011)

OLIVETTI M10 CON SCHEDA ESTERNA COMPLETA DI
CONVERT. A/D A 4 CANALI, CONV. D/A, BUFFER PER 8 RELÈ
DI USCITA. ADATTO AD ACQUISIRE DATI E AD INVIARE
COMANDI, VENDO SOLTANTO IN BLOCCO CON SCHEMI
A LIRE 500'000. TEL. 040-422231 ORE SERALI.

CERCO SOFTWARE PER C128 ANCHE IN
MODALITÀ CP/M. SCAMBIO CON SOFTWARE
PER C64 E C128.
MASSIMILIANO VIOLANTE, VIA C. PORZIO, 106
80139 - NAPOLI

POSTAMIGA

(a cura di R. F.)

DANNATO GURU!

☐ Con il mio Amiga 500, caricando un videogame, appare sul video il messaggio:
"Recoverable Alert. Press left mouse button to continue. Guru Meditation 30000001.00000004"
Ho cercato di recuparare il programma ma non ci sono riuscito. Come posso fare? Perchè non parlate dei programmi inclusi nel disco Extras?
(Zullo Domenicantonio - Alvignano)

• Dalla descrizione dei messaggi inviati dal diskdoctor mi pare di capire che il videogame non è in versione originale, ma pirateggiata: nel processo di spotezione è stata modificata la struttura del disco, con l'inserimento addirittura dell'AmigaBasic (ma come ci è finito?). Non è difficile supporre che il videogame Xenon (che peraltro è molto protetto) ha mal sopportato questa intrusione, ritrovandosi magari alcuni file di dati danneggiati.

Insomma un disco sprotetto (soprattutto con Amiga) è un disco molto delicato perchè anche se si è riusciti ad aggirare la protezione, può bastare ancora una minima modifica per rovinare tutto. Il messaggio di Guru è il minimo che possa capitare in casi come questo; considera che esistono programmi che, accortisi di girare in copia, cominciano a sbattere rapidamente la testina del drive con l'intento di disallinearla (e con le conseguenze che puoi immaginare). Un disco sprotetto, che "va" in read / write error o cose del genere, nel 90% dei casi non può più essere recuperato. Non resta che rassegnarsi e procurarsi un'altra copia (magari originale...).

Circa il secondo quesito, la directory "Extras" contiene, in maggior parte, programmi dimostrativi in AmigaBasic, file di tipo ".fd" per la creazione dei file ".bmap" (utili all'AmigaBasic per richiamare le funzioni del Kernel), ed una directory di Tools. Tra questi spicca il text editor Micro Emacs, molto adatto per la stesura di sorgenti di programma. Purtroppo comprende molte opzioni e non è questa la sede adatta per illustrarle tutte.

TASTIERE

☐ Ho notato che sono in circolazione due tipi di tastiera per l'Amiga. Una ha dei tasti in più dell'altra ed il Return più piccolo. Che differenza c'è tra i due modelli?
(Mauro Bricca - Diano Marina)

• Abbiamo già affrontato su queste pagine il problema delle diverse tastiere di Amiga. La differenza tra i due modelli non è, come ipotizzi, una differenza di età, ma di nazionalità. La tastiera con il tasto Return più grande (e senza le vocali accentate) è la tastiera americana, mentre quella con il return piccolo e le vocali accentate è quella italiana. La Commodore ha infatti previsto la nazionalizzazione della tastiera di Amiga e quindi esistono tastiere inglesi, tedesche, svedesi, italiane... a seconda del Paese in cui il computer viene messo in vendita.

Ti posso assicurare che le differenze non provocano alcun problema di incompatibilità di programmi. Al massimo può succedere che il computer risponda seguendo lo standard americano, invece che italiano, ma anche questo è improbabile visto che nella startup-sequence (procedura di avvio) di ogni Workbench italiano è inserito un apposito comando che impone al computer di riconoscere la tastiera italiana (si tratta di "SetMap I").

ARSENICO E NUOVI COMPUTER

☐ E' vero che i virus possono danneggiare irrimediabilmente l'eventuale

hard disk collegato ad Amiga? Conviene sostituire il 68000 dell'Amiga 2000 con una scheda dotata di 68020? Vorrei chiedervi informazioni sul nuovo Amiga 2500 ed il relativo ambiente Unix.

(Tressi Luca e Della Gioia Nicola)

• Purtroppo è vero. Uno degli obiettivi preferiti dai virus è l'hard disk. Naturalmente non tutti i virus sono così cattivi, soprattutto nel caso di Amiga, ma questi infernali programmi fioriscono di giorno in giorno ed è difficile dire quali attaccano l'hard disk e quali no.

Una buona notizia per i possessori di Hard disk viene comunque dalla versione 1.3 del Sistema Operativo. Il nuovo Kickstart, infatti, prevede, tra i dispositivi di boot, anche l'hard disk. Non sarà cioè più necessario inserire il Workbench all'accensione della macchina se questo è già presente sull'hard disk.

In questo modo si sbarra la strada ai virus che riescono ad installarsi in memoria solo a patto di trovarsi sul disco con cui si fa il boot (avvio) di Amiga.

Purtroppo è sempre possibile che si sia costretti a lanciare, da disco, un programma, permettendo l'installazione in memoria di un eventuale virus. La soluzione di rifornirsi solo di software originale non sembra essere molto pratica sia per motivi economici (saremmo rapidamente costretti a chiedere un mutuo pluriennale), sia perchè, volenti o nolenti, una biblioteca di software è fatta di tanti programmi di varia provenienza (programmi di Pubblico Dominio, Utility elaborate in proprio e scambiate tra amici...) ed il virus possiamo averlo preso in tantissime maniere.

L'uso di un buon antivirus sembra, per adesso, essere la soluzione più ovvia ed adatta. Sarà seccante dover scrutare e controllare ogni dischetto "straniero" che ci giunge, ma senz'altro molto sicuro.

Nei comandi del CLI v1.3 (lo Shell) è stato a questo scopo incluso il comando...
INSTALL DFn: CHECK
...che verifica la bontà del bootblock del disco inserito nel drive n.

La sostituzione del microprocessore Amiga è una questione molto complessa. In linea di massima le schede acceleratrici non sono un buon affare in termini economici (leggi: sono costose), anche se effettivamente possono dare incredibili guadagni in termini di velocità, soprattutto se accompagnati da una memoria adeguata a 32 bit.

Recentemente la Commodore ha però presentato (ma non ancora distribuito) una scheda con 68020 e coprocessore matematico per l'utilizzo di Unix su Amiga 2000 (che così equipaggiato dovrebbe chiamarsi 2500). In questa prospettiva le cose cambiano perchè l'alto prezzo potrebbe essere giustificato dalla possibilità di adoperare un sistema operativo come Unix che gira solo su computer ben più costosi di un Amiga. Il discorso è comunque molto articolato e vedremo di riparlarne quanto prima, vagliando tutte le possibilità e soluzioni dell'alternativa 68020/30.

DA C/128 A...

□ Posseggo un C/128 ma sento il bisogno di passare ad un sistema superiore come Amiga 500. Sarei intenzionato ad acquistare la tastiera, il monitor e il secondo disk drive. Oltre a questi sono indeciso se acquistare l'espansione di memoria oppure il C/64 Emulator. Cosa mi consigliate? Che cosa significano i termini: monitor ad alta persistenza, monitor antiriflesso, filtro antiriflesso, sfarfallamento? Ho saputo che l'Amiga 1000 è uscito di produzione perchè sono stati riscontrati sovente guasti hardware. Si dice che la stessa cosa stia accadendo all'Amiga 500. E' vero?

(Lo Giacco Roberto)

• L'espansione di memoria è un accessorio molto utile ed è praticamente indispensabile se non vuoi limitarti ad usare solo giochi (anche quelli comunque cominciano a stare stretti in 512 K, vedi Dragon's Lair). Purtroppo questo è un brutto momento per velleità mnemoniche. Data la carenza di chip RAM, verificatasi sul mercato, tutte le espansioni hanno cominciato a far lievitare i loro prezzi, fino anche a raddoppiarsi. Se non ne hai immediato bisogno, puoi provare ad aspettare qualche tempo e sperare in un ridimensionamento dei prezzi. Sul C/64 Emulator non ci dilunghiamo se non per consigliarti di rileggere la prova apparsa il numero scorso e trarre le dovute conclusioni.

Lo sfarfallamento del video di Amiga (in inglese "flickering") si verifica quando usiamo le risoluzioni di 320 X 512 oppure 640 X 512. Il supporto di queste risoluzioni, infatti, in Amiga avviene attraverso il modo "interlacciato" cioè con doppia scansione raster del video che provoca un dimezzamento dei tempi di refresh di un intero quadro, con il risultato di un leggero tremolio dell'immagine (il flickering, appunto).

Per rimediare a questo difetto esistono molti modi. Il più semplice, economico ed immediato è quello di regolare opportunamente contrasto e luminosità del monitor. Scurendo leggermente l'immagine, e riducendo il contrasto, si assiste ad una netta diminuzione del flickering. Naturalmente bisognerà trovare un giusto compromesso tra contrasto e sfarfallamento: non serve a nulla eliminare lo sfarfallamento se non riusciamo più a vedere cosa compare sul video!

Un altro metodo è quello di piazzare davanti allo schermo un pannello di plastica trasparente offuscata, che si trova in commercio ad un prezzo intorno alle 40000 lire (Jitter-Rid), ma ha circa lo stesso effetto

di ridurre semplicemente la luminosità. Anche la Commodore ha proposto un rimedio: un monitor ad alta persistenza denominato 2080. La sua azione è quella di trattenere più a lungo l'immagine sul video attenuando il fastidioso effetto. Non è una soluzione drastica (il flickering rimane ancora, pur se attenuato leggermente) e può presentare inconvenienti in giochi tutta azione spara e fuggi (proiettili che lasciano la scia sul monitor...). Il costo del monitor è comunque abbastanza contenuto, considerando la qualità, ed è indicato per chi fa un intenso uso della grafica in modo interlacciato.

La soluzione migliore è invece quella di dotarsi di un accessorio per Amiga 2000 chiamato flickerFixer prodotto dalla MicroWay, e di un monitor ad alta risoluzione (NEC Multisync o cose del genere). Collegando il flickerFixer all'Amiga si ottiene una uscita video non interlacciata, che può essere inviata ad un monitor in grado di supportare una risoluzione di almeno 640 X 512 pixel. Questa soluzione presenta un solo inconveniente: è costosissima. Il solo flickerFixer costa una cifra intorno alle 900000 lire a cui va aggiunto il costo del monitor (1000 - 1500 lire "pesanti") visto che quello fornito dalla Commodore (il 1084) non riesce a supportare una risoluzione di 512 punti verticali non interlacciati.

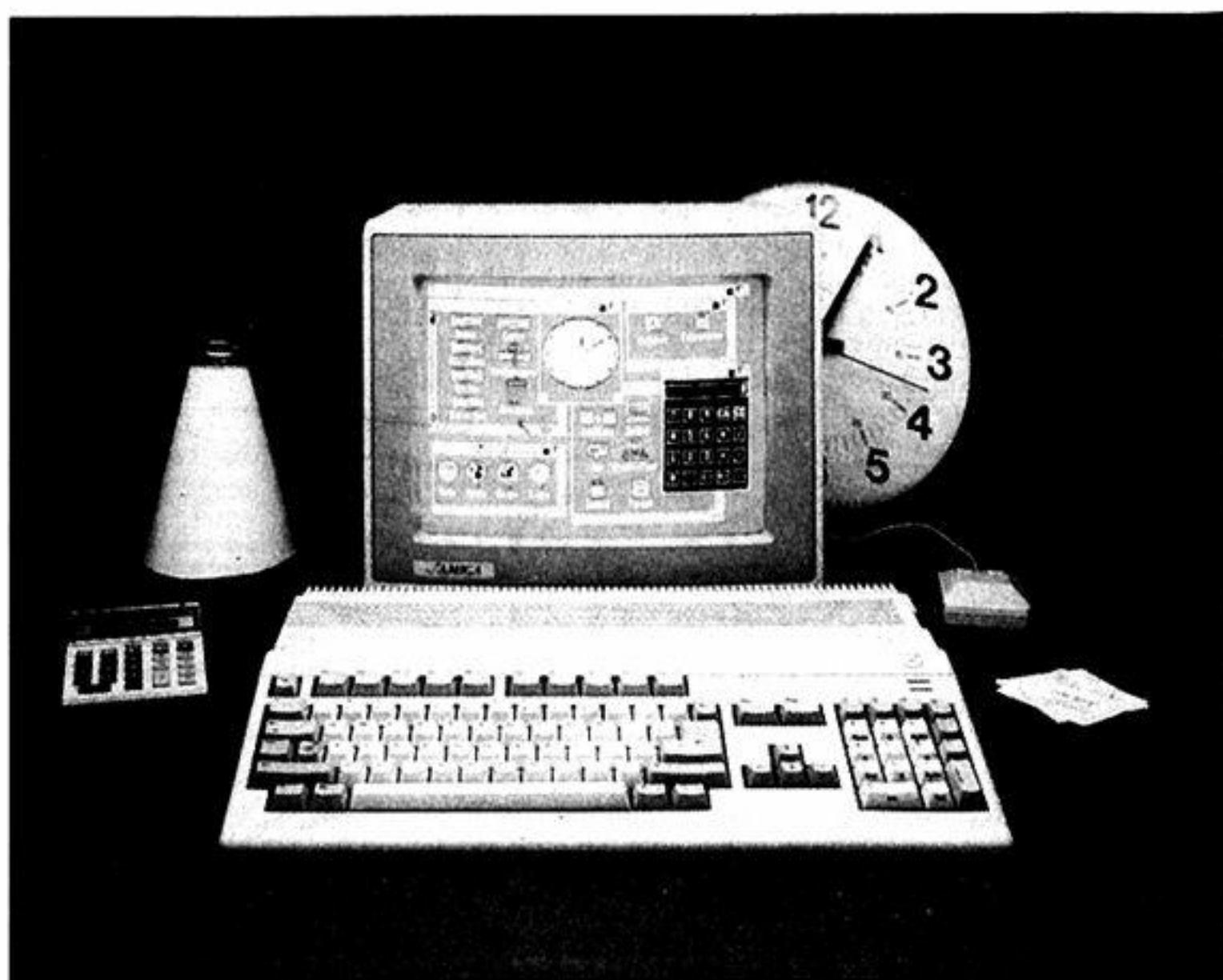
Come vedi ci sono diverse soluzioni per tutte le tasche, a te la scelta.

Effettivamente il modello 1000 presentava qualche pecca dovuta non tanto al progetto del computer, quanto alla sua realizzazione. I nuovi modelli, checchè ne dicano gli invidiosi, funzionano perfettamente, e quand'anche si dovesse verificare un malfunzionamento, ricorda che la Commodore Italiana garantisce i suoi prodotti per un anno.

TRA AMIGHE

□ E' vero che tutti i modelli Amiga sono compatibili tra loro? Conviene procurarsi un Amiga visto che tutti i suoi giochi non sono altro che conversioni da Atari ST?

(Filippo Piscopo - Messina)

• I modelli 500 e 2000, a parità di configurazione (memoria, disk drives) sono perfettamente compatibili. Il 2000 ha il vantaggio di poter sfruttare i programmi in ambiente MS-DOS, OS/2 e Unix grazie alle schede che via via la Commodore realizzerà. Le uniche incompatibilità si trovano a livello hardware. Da questo punto di vista, molti accessori per il 1000 non vanno bene per il 500, e viceversa, a causa delle strane interfacce di cui è dotato il modello 1000. Altre incompatibilità si possono riscontrare con gli hard disk oppure con i dispositivi genlock, ma si tratta spesso più di problemi meccanici che di connessione. I migliori rivenditori dispongono, tuttavia, di accessori idonei per ognuno dei tre modelli Amiga 1000, 500 e 2000.

Sulla questione dei giochi bisogna innanzitutto dire che non è affatto vero che TUTTI i giochi per Amiga sono conversioni da ST (i più belli sono stati scritti sull'Amiga, credi a me...), anche se questo vale per una parte consistente della produzione videogamica. Tieni comunque presente che, almeno da un punto di vista sonoro, i giochi per Amiga sono infinitamente superiori a quelli di un ST. Ti invito a considerare l'eventuale acquisto di un Amiga non solo come macchina per giocare, ma anche e soprattutto come computer che ti sia di aiuto nello svolgere tanti altri compiti (studio, lavoro, apprendimento della programmazione). In questo modo eviterai di accantonare il nuovo acquisto non appena comincerai ad essere stanco di giocare.

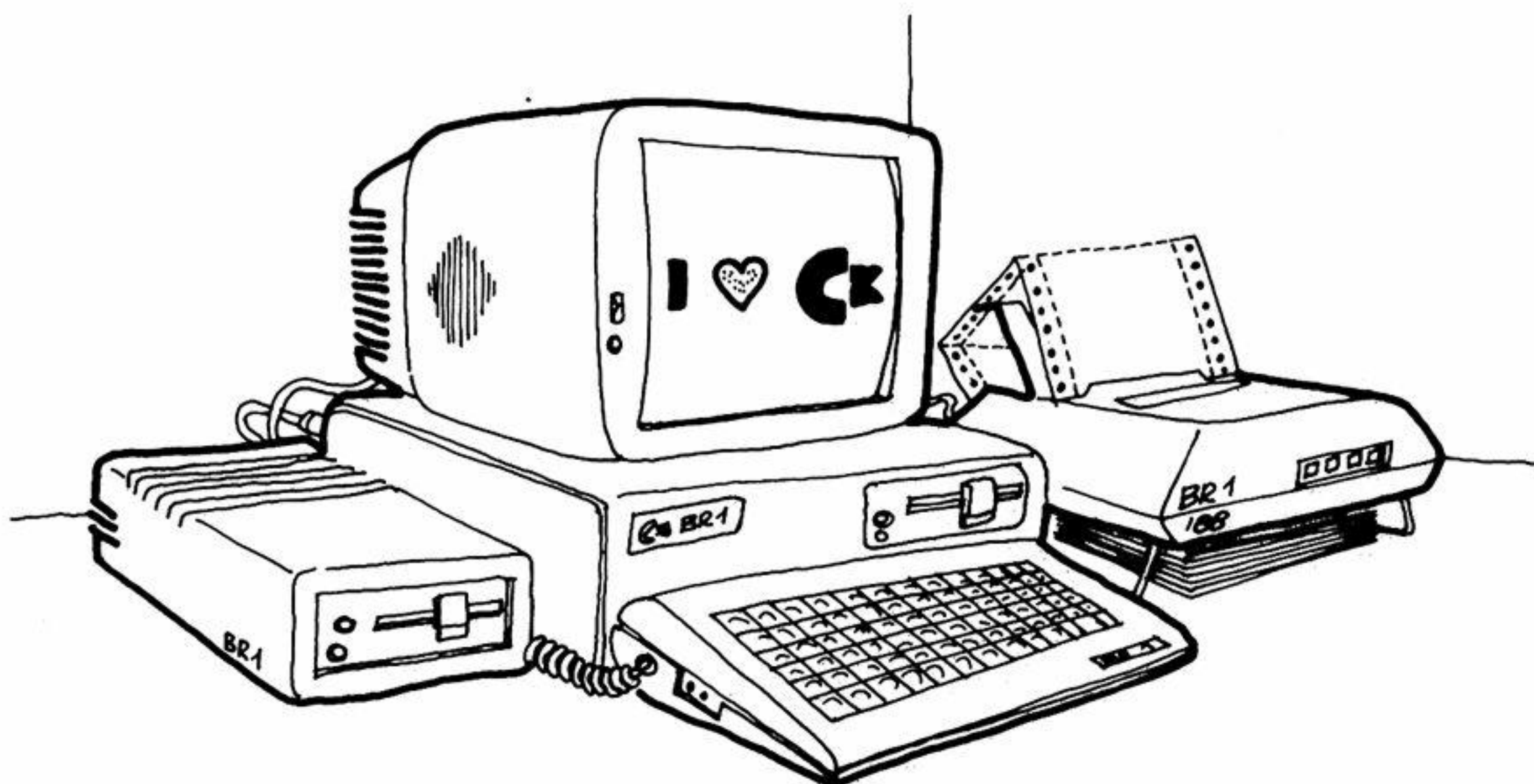
LE LUCI DI AMIGA

□ Ho notato che alcuni programmi determinano lo spegnimento del led di Amiga contrassegnato "power", apparentemente senza ricavarne alcun beneficio. L'unica variazione rilevabile è un certo cambiamento nella tonalità dei suoni (più metallici e fastidiosi). Quali variazioni nel computer comporta questo fatto?

(Marco Tendas - Roma)

• Nessuna, oltre a quella da te rilevata. Amiga è in grado di riprodurre suoni con frequenza fino a 15000 Hertz, ma in uscita arrivano solo quelli fino a 10000 circa.

Questo succede perchè all'uscita dei suoni c'è un filtro che permette solo alle frequenze più basse di 4000 Hertz di passare, e per questo viene denominato filtro passa - basso. Tutti i suoni con frequenza maggiore vengono via via smorzati, fino a non sentirli più per frequenze di 8000 - 10000 Hertz. Questo filtro era costantemente attivo negli Amiga 1000. Negli Amiga 500 e negli Amiga 2000, invece, è disattivabile.

La sua esclusione è demandata allo stesso bit che controlla la luminosità del led di accensione (che non si spegne, ma dimezza la sua luminosita). Alcuni programmi musicali prevedono una apposita opzione per la disattivazione del filtro in questione. Naturalmente ha effetto solo sugli Amiga 500 e 2000, risultando inutile sul 1000. I suoni, pertanto, non diventano "più metallici": diventano udibili frequenze che normalmente vengono attenuate, oppure del tutto eliminate.

COME TI STAMPO LO SPRITE MULTICOLOR

Riportare su carta uno sprite non è semplice come riportare una schermata hi-res; soprattutto se l'immagine è in multicolor

di **Marco Corazza**

La routine che proponiamo stavolta è la logica continuazione di un discorso intrapreso alcuni mesi fa (CCC n. 54), relativo alla stampa degli sprite "normali".

Con la routine pubblicata in precedenza potevamo stampare sprite solo se definiti in modo monocolore; ora, invece, possiamo stampare anche quelli multicolore grazie ad un artificio di programmazione da realizzare, inutile dirlo, in linguaggio macchina.

Gli sprite multicolor (lo ricordiamo ai lettori distratti) hanno la caratteristica di essere più colorati di quelli normali (sono ben quattro i colori utilizzabili) ma, come contropartita, perdono in risoluzione orizzontale. In parole povere, invece di una griglia di 24 punti orizzontali e 21 verticali abbiamo a disposizione una griglia di soli 12 punti orizzontali; quelli verticali rimangono sempre 21.

Il motivo della "riduzione" orizzontale è piuttosto semplice da comprendere: mentre, in condizioni standard, il computer assegna ad ogni punto un bit (0 = spento; 1 = acceso) nel caso del multicolor ha bisogno di 2 bit per ciascun punto. A seconda della combinazione di tali 2 bit (cioè: 00, 01, 10 oppure 11) il computer assegna, poi, un colore in accordo con la tabella 1 riportata in queste pagine.

Per settare i colori a nostro piacimento, è sufficiente agire sui registri indicati nella tabella, ricordando che sfondo, multicolor n. 0 e multicolor n. 1 sono fissi per tutti gli sprite.

Ora che sappiamo tutto in materia, passiamo alla spiegazione della routine, che non è poi così complessa. Il programma scorre, uno per volta, i 63 byte dello sprite indicato, utilizzando un contatore di righe ed uno di colonne.

Ogni byte viene shiftato (spostato) di 2 bit per quattro volte ed ogni volta, a seconda della combinazione di bit, viene selezionato uno dei 4 colori possibili. Per ogni colore il programma ha in memoria la definizione di una griglia di 7 x 7 pixel, che viene quindi mandata in stampa. Usando il procedimento descritto, la riproduzione su carta risulta molto ingrandita rispetto alla stampa di uno sprite monocolore; non avrete più bisogno della lente d'ingrandimento!

LA SINTASSI

La sintassi da usare è la seguente:

Sys XXXX, Spr, C1, C2, C3, C4 [, Banco [, Device]]

Bit	Colore	Registro
00	sfondo	53281
01	multicol #0	53285
10	col. sprite	53287-53294
11	multicol #1	53286

Tabella 1.

Banco	Locazioni
0	0 - 16383
1	16384 - 32767
2	32768 - 49151
3	49152 - 65535

Tabella 2.

...in cui:

- XXXX è l'indirizzo della routine (qualsiasi, dal momento che è rilocabile).
- Spr è il numero dello sprite da stampare (tra 0 e 255).
- C1, C2, C3, C4 sono i colori, come da tabella 1.
- Banco e Device sono opzionali e, se omessi, valgono 0 per il Banco e 4 per il dispositivo (cioè la stampante).

Da notare che, in assenza dei parametri relativi al colore, il programma prenderà, per default, i contenuti dei registri corrispondenti (53287 per C3).

Per individuare la locazione d'inizio, in cui è situato lo sprite, bisogna usare la formula:

Loc = 16384 x Banco + 64 x Sprite

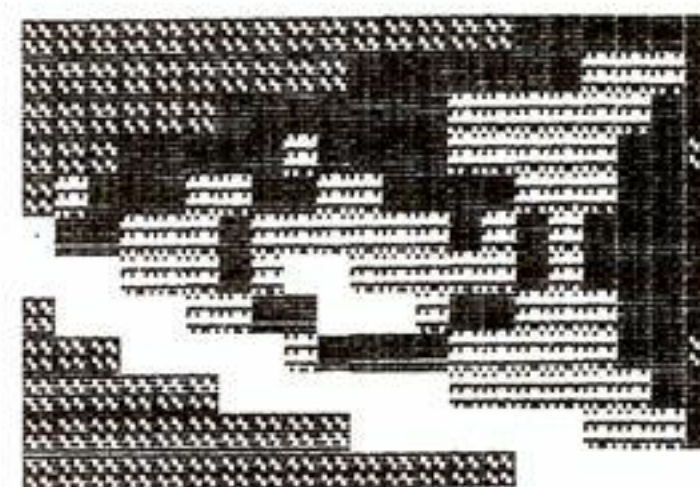
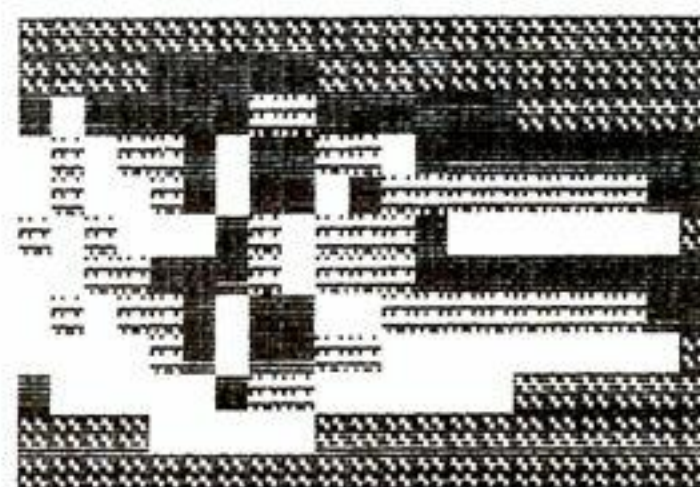
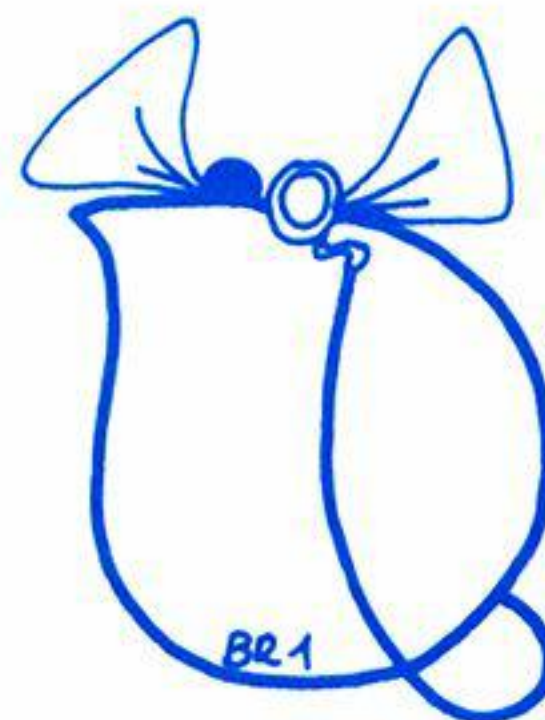
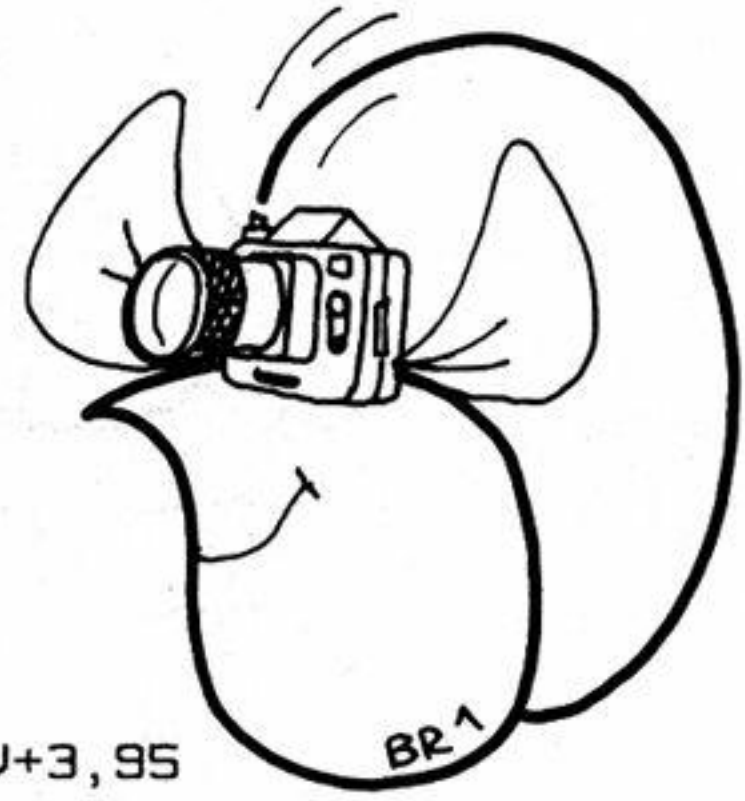
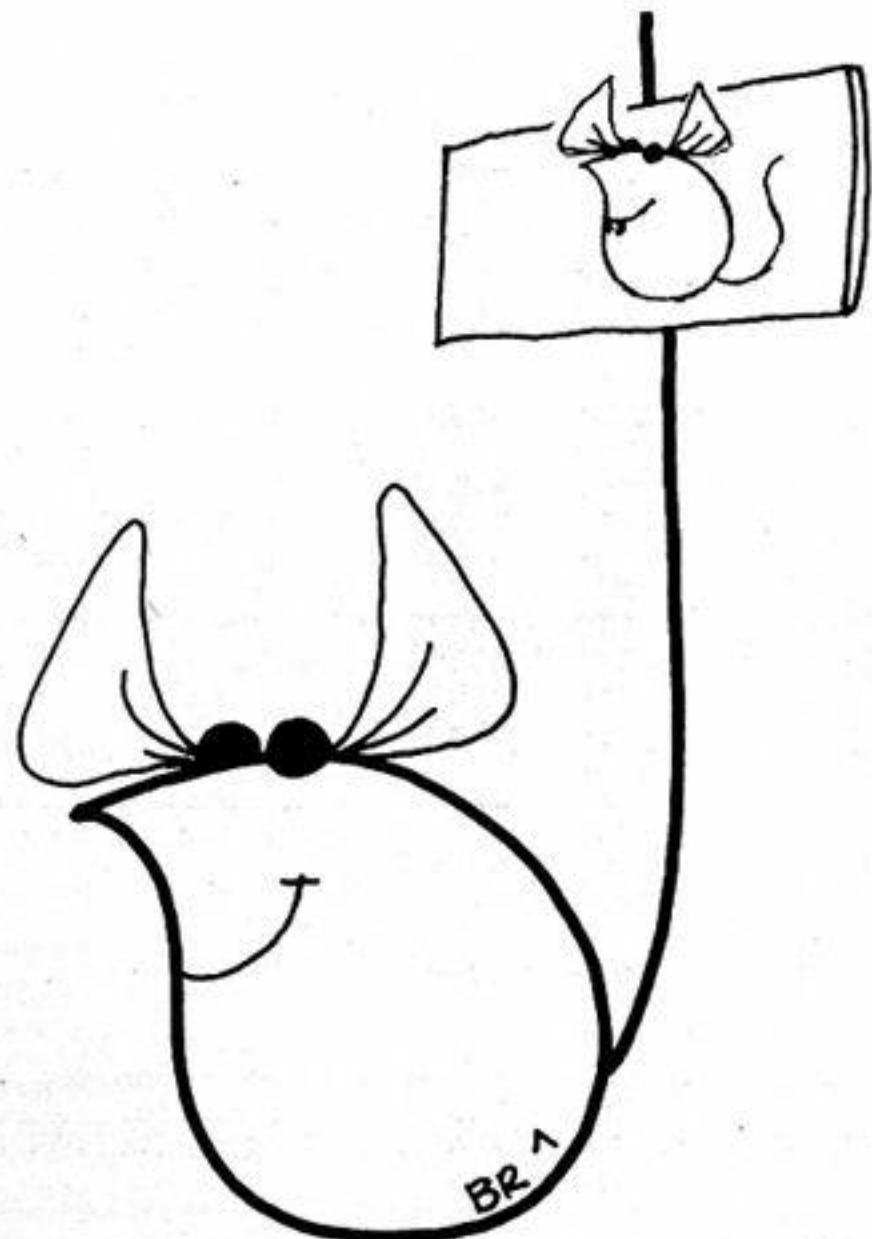
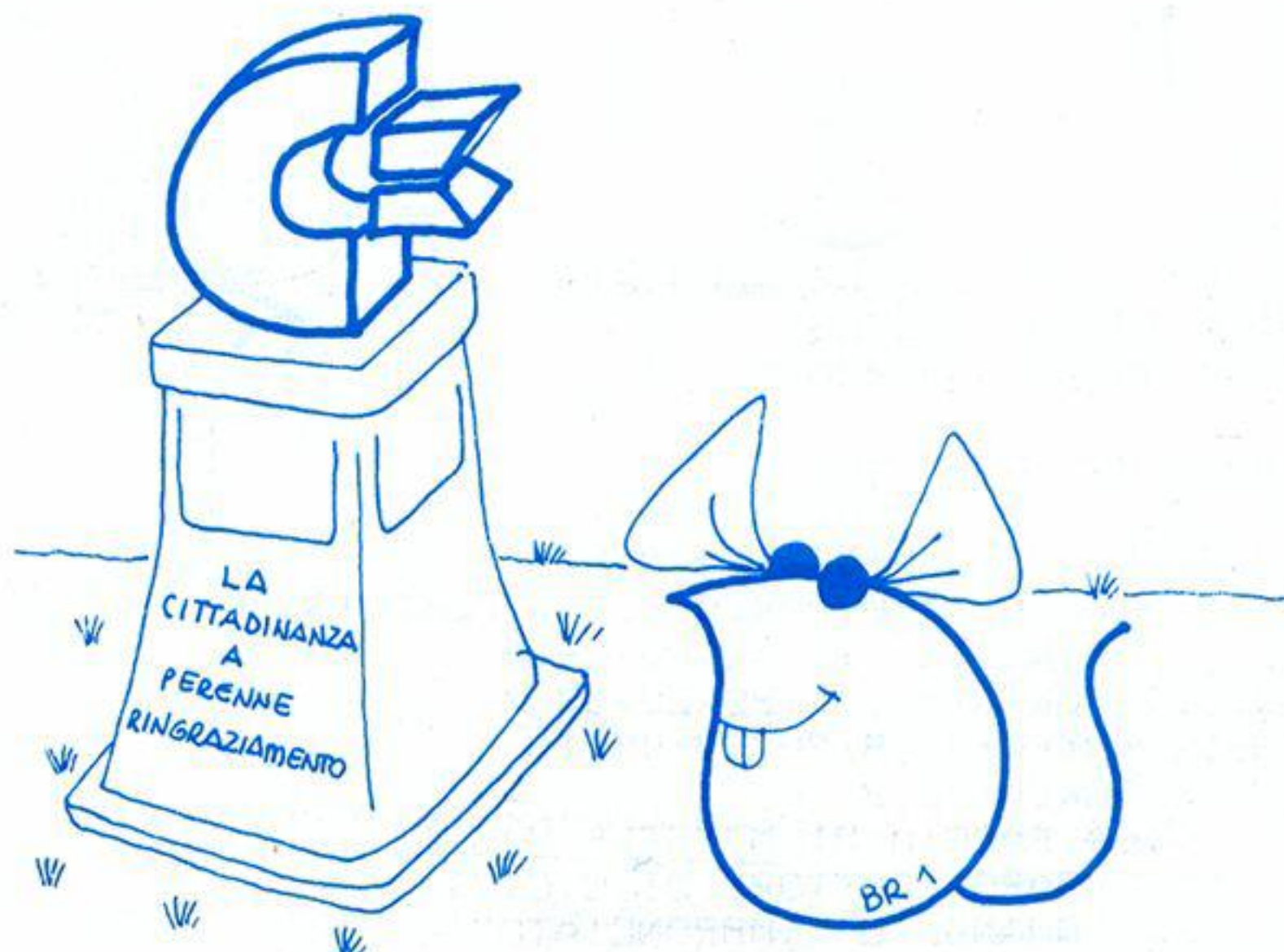
Il parametro Banco permette di selezionare uno dei 4 segmenti di memoria del C/64, come risulta, del resto, anche dalla tabella 2.

L'ultima raccomandazione è quella di non usare le locazioni da $F7 a $FC e da $0334 a $033D perchè vengono utilizzate dalla routine.

Per coloro che volessero provare la stampa, senza perdere tempo a disegnare ed allocare sprite, abbiamo preparato un mini programma Basic dimostrativo.

Buon divertimento!

```
1000 PRINTCHR$(147);"STAMPA SPRITES MULTICOLORI":PRINT
1010 PRINT"SINTASSI:":PRINT
1020 PRINT"SYS XXXX,SPR,C1,C2,C3,C4 [,BANK [,DEV]]":PRINT:PRINT
1030 PRINT"XXXX=INDIRIZZO DI PARTENZA QUALSIASI":PRINT
1040 PRINT"SPR=NUMERO SPRITE (0..255)":PRINT
1050 PRINT"C1=COLORE SFONDO  DEFAULT=REG.53281"
1060 PRINT"C2=MULTICOLOR #0  DEFAULT=REG.53285"
1070 PRINT"C3=COLORE SPRITE  DEFAULT=REG.53287"
1080 PRINT"C4=MULTICOLOR #1  DEFAULT=REG.53286":PRINT
1090 PRINT"BANK=BANCO (0..3) DEFAULT=0":PRINT
1100 PRINT"DEV=PERIFERICA    DEFAULT=4"
1110 RETURN
1200 :
1209 DATA 169,000,133,247,141,061,003
1219 DATA 169,004,133,248,173,033,208
1229 DATA 041,015,141,052,003,173,037
1239 DATA 208,041,015,141,053,003,173
1249 DATA 039,208,041,015,141,054,003
1259 DATA 173,038,208,041,015,141,055
1269 DATA 003,024,165,020,105,078,141
1279 DATA 006,129,165,021,105,001,141
1289 DATA 007,129,032,253,174,032,158
1299 DATA 183,169,000,133,250,134,249
1309 DATA 160,006,006,249,038,250,136
1319 DATA 208,249,169,044,160,000,209
1329 DATA 122,208,067,032,253,174,032
1339 DATA 158,183,224,016,144,003,076
1349 DATA 072,178,172,061,003,138,153
1359 DATA 052,003,238,061,003,200,192
```

```
1369 DATA 004,208,220,169,044,160,000
1379 DATA 209,122,208,031,032,253,174
1389 DATA 032,158,183,224,004,144,003
1399 DATA 076,072,178,134,247,169,044
1409 DATA 160,000,209,122,208,008,032
1419 DATA 253,174,032,158,183,134,248
1429 DATA 165,249,133,251,165,247,010
1439 DATA 010,010,010,010,010,024,101
1449 DATA 250,133,252,169,004,166,248
1459 DATA 160,000,032,186,255,169,000
1469 DATA 032,189,255,032,192,255,162
1479 DATA 004,032,201,255,169,008,032
1489 DATA 210,255,169,000,141,059,003
1499 DATA 141,060,003,160,000,177,251
1509 DATA 162,004,141,056,003,142,057
1519 DATA 003,140,058,003,010,144,013
1529 DATA 010,144,005,172,055,003,016
1539 DATA 016,172,054,003,016,011,010
1549 DATA 144,005,172,053,003,016,003
1559 DATA 172,052,003,240,009,024,169
1569 DATA 000,105,007,136,208,251,168
1579 DATA 162,007,185,078,129,032,210
1589 DATA 255,200,202,208,246,173,056
1599 DATA 003,174,057,003,172,058,003
1609 DATA 010,010,202,208,183,230,251
1619 DATA 208,002,230,252,238,060,003
1629 DATA 173,060,003,201,003,208,161
1639 DATA 169,010,032,210,255,169,000
1649 DATA 141,060,003,238,059,003,173
1659 DATA 059,003,201,021,208,141,169
1669 DATA 013,032,210,255,169,015,032
1679 DATA 210,255,076,231,255,255,255
1689 DATA 255,255,255,255,255,128,128
1699 DATA 128,128,128,128,128,146,129
1709 DATA 236,218,129,236,218,146,128
1719 DATA 164,201,128,164,201,218,218
1729 DATA 165,165,218,165,165,201,255
1739 DATA 201,255,255,201,255,219,255
1749 DATA 164,182,255,164,182,146,183
1759 DATA 200,164,183,200,164,218,255
1769 DATA 128,165,255,128,165,254,237
1779 DATA 183,219,237,183,219,200,146
1789 DATA 128,164,146,128,164,255,219
1799 DATA 237,255,219,237,255,219,128
1809 DATA 200,128,128,200,128,129,164
1819 DATA 146,137,164,146,137,200,129
1829 DATA 164,129,129,164,129,128,128
1839 DATA 164,200,128,164,200,-1,58169
1840 END
```

```
100 REM ** DEMO CON ROUTINE ALLOCATA
110 REM ** DA $8000 = 32768
120 REM ** BY MARCO CORAZZA
130 :
140 PRINTCHR$(147):V=53248
150 FORI=832TO894:READA:POKEI,A:NEXT
160 FORI=896TO958:READA:POKEI,A:NEXT
170 POKEV,150:POKEV+1,200:POKEV+2,140:POKEV+3,95
180 POKE2040,13:POKE2041,14:POKEV+28,3
190 POKEV+32,11:POKEV+33,11:POKEV+37,0
195 POKEV+38,1:POKEV+39,8:POKEV+40,13
200 POKEV+23,3:POKEV+29,2
210 PRINT"      LA FACCIA DI MICHELE MAGGI"
220 PRINT"       (CAPOREDATTORE DI C.C.C)"
230 PRINT"        QUANDO E' IMBRONCIATO"
240 FORI=1TO9:PRINT:NEXT
250 PRINT"       L'ASTRONAVE PER SFUGGIRE"
260 PRINT"             ALLE IRE DI"
270 PRINT"        MICHELE MAGGI INFURIATO"
300 :
310 POKEV+21,3
350 POKE198,0:WAIT198,1:POKE198,0
370 POKEV+21,0
380      SYS 32768,13: SYS 32768,14
400 END
1009 DATA 000,060,000,000,246,000,000
1019 DATA 245,000,003,233,064,003,233
1029 DATA 064,003,170,064,015,150,080
1039 DATA 015,105,080,014,121,144,013
1049 DATA 250,080,061,234,084,061,233
1059 DATA 084,061,169,084,058,101,164
1069 DATA 058,105,164,250,150,165,250
1079 DATA 170,165,234,150,169,233,085
1089 DATA 105,229,085,089,212,085,021
1100 REM
1109 DATA 007,251,208,015,190,176,015
1119 DATA 235,208,015,175,144,062,158
1129 DATA 148,061,093,084,055,215,212
1139 DATA 057,105,100,057,125,100,014
1149 DATA 235,144,014,233,144,015,170
1159 DATA 208,015,150,080,015,158,080
1169 DATA 015,158,080,003,158,064,003
1179 DATA 158,064,003,158,064,003,158
1189 DATA 064,003,093,064,000,081,064
```

```
1 ******** STAMPA-SPR2.3 *********
2 *                              *
3 * STAMPA SPRITE MULTICOLORI    *
4 *                              *
5 *   BY: MARCO CORAZZA          *
6 *                              *
7 *        OTTOBRE '88           *
8 *                              *
9 ********************************
10
11 INIZIO  LDA   #00        ;assegna valori
12         STA   $F7        ;di default
13         STA   CONT
14         LDA   #04
15         STA   $F8        ;BANCO = 0
16         LDA   53281      ;DEVICE= 4
17         AND   #15
18         STA   $0334      ;C1 - sfondo
19         LDA   53285      ;C2 - multi #0
20         AND   #15
21         STA   $0335      ;C3 - colore
22         LDA   53287      ;       sprite 1
23         AND   #15
24         STA   $0336      ;C4 - multi #1
25         LDA   53286
26         AND   #15
27         STA   $0337
28         CLC
29         LDA   $14        ;setta indirizzo
30         ADC   #$4E       ;tabella-colori
31         STA   O3+1
32         LDA   $15
33         ADC   #$01
34         STA   O3+2
35 *
36         JSR   SALVIR     ;controlla 1' virgola
37         JSR   GETVAL     ;e legge SPR poi lo
38         LDA   #00        ;moltiplica per 64 e
39         STA   $FA        ;lo salva in $F9/$FA.
40         STX   $F9
41         LDY   #06
42 MOLTIP  ASL   $F9
43         ROL   $FA
44         DEY
45         BNE   MOLTIP
46 *
47 NEXT    LDA   #$2C       ;acquisisce i valori
48         LDY   #$00       ;dei 4 colori.
49         CMP   ($7A),Y    ;in assenza di para-
50         BNE   START      ;metri salta a START
51         JSR   SALVIR
52         JSR   GETVAL
53         CPX   #16        ;se parametri >= 16
54         BCC   OKVAL      ;salta a
55         JMP   ILQUER     ;ILLEGAL QUANTITY
56 OKVAL   LDY   CONT       ;ERROR
57         TXA
58         STA   $0334,Y
59         INC   CONT
60         INY
61         CPY   #04
62         BNE   NEXT
63 *
64         LDA   #$2C       ;controlla la vir-
65         LDY   #$00       ;gola, se c'e'
66         CMP   ($7A),Y    ;acquisisce il
67         BNE   START      ;valore del BANCO
68         JSR   SALVIR
69         JSR   GETVAL     ;se e' >= 4
70         CPX   #04        ;salta a
71         BCC   BANCO      ;ILLEGAL QUANTITY
72         JMP   ILQUER     ;ERROR.
73 BANCO   STX   $F7
74         LDA   #$2C       ;controlla l'ulti-
75         LDY   #$00       ;ma virgola, se
76         CMP   ($7A),Y    ;c'e' legge il
77         BNE   START      ;DEVICE.
78         JSR   SALVIR
79         JSR   GETVAL
80         STX   $F8
81 *
82 START   LDA   $F9        ;calcola indirizzo
83         STA   $FB        ;iniziale dei dati
84         LDA   $F7        ;da stampare:
85         ASL   A          ;INDIR = BANCO *
86         ASL   A          ;16384 + SPR * 64
```

```
87            ASL   A
88            ASL   A
89            ASL   A
90            ASL   A
91            CLC
92            ADC   $FA
93            STA   $FC
94 *
95            LDA   #$04         ;apre il canale 4
96            LDX   $F8          ;e DEVICE scelto
97            LDY   #$00
98            JSR   SETLFS
99            LDA   #$00
100           JSR   SETNAM
101           JSR   OPEN
102           LDX   #$04
103           JSR   CHKOUT       ;seleziona il modo
104           LDA   #$08         ;stampa immagine
105           JSR   CHROUT       ;bit a bit
106 *
107           LDA   #$00         ;inizializza i
108           STA   R            ;contatori R-righe
109           STA   C            ;e C-colonne
110 FOR       LDY   #00          ;preleva il byte
111           LDA   ($FB),Y      ;da trattare e lo
112           LDX   #04          ;shifta a sinistra
113 SHIFT     STA   SA           ;di 2 bit
114           STX   SX           ;salva i registri
115           STY   SY
116           ASL   A
117           BCC   Z1           ;in base alla coppia
118 U1        ASL   A            ;di bit incontrata
119           BCC   U2           ;pone in Y il colore
120 UU        LDY   $0337        ;e va alla stampa
121           BPL   OUTDATI
122 U2        LDY   $0336
123           BPL   OUTDATI
124 Z1        ASL   A
125           BCC   Z2
126 ZU        LDY   $0335
127           BPL   OUTDATI
128 Z2        LDY   $0334
129 *
130 OUTDATI   BEQ   O2           ;esegue A=A+7 per Y
131           CLC                ;volte
132           LDA   #00          ;posizionandosi sui
133 O1        ADC   #07          ;dati del colore
134           DEY                ;prescelto
135           BNE   O1
136           TAY                ;usa Y come indice
137 O2        LDX   #07
138 O3        LDA   TABCOL,Y
139           JSR   CHROUT       ;stampa matrice
140           INY                ;7*7 del colore
141           DEX                ;selezionato
142           BNE   O3
143 *
144           LDA   SA .         ;ripristina i
145           LDX   SX           ;registri
146           LDY   SY
147           ASL   A            ;shifta ancora A
148           ASL   A
149           DEX                ;se x=0 prosegue
150           BNE   SHIFT        ;altrimenti torna
151 *                            ;su
152           INC   $FB          ;INDIR=INDIR+1
153           BNE   COLON
154           INC   $FC
155 COLON     INC   C            ;incrementa C
156           LDA   C            ;se < 3 torna su
157           CMP   #03
158           BNE   FOR
159 *
160 RIGA      LDA   #10          ;altrimenti azzera C,
161           JSR   CHROUT       ;manda a capo la
162           LDA   #00          ;stampante
163           STA   C
164           INC   R            ;incrementa R.
165           LDA   R            ;se R < 21
166           CMP   #21          ;torna su
```

```
167           BNE   FOR
168 *
169 END       LDA   #13          ;altrimenti invia
170           JSR   CHROUT       ;CHR$(15) , chiude
171           LDA   #15          ;tutti i canali
172           JSR   CHROUT       ;aperti e torna
173           JMP   CLALL        ;al BASIC.
174 *
175 * DEFINIZIONE COLORI
176 *
177 TABCOL    DFB   255,255,255,255,255,255,255
178           DFB   128,128,128,128,128,128,128
179           DFB   146,129,236,218,129,236,218
180           DFB   146,128,164,201,128,164,201
181           DFB   218,218,165,165,218,165,165
182           DFB   201,255,201,255,255,201,255
183           DFB   219,255,164,182,255,164,182
184           DFB   146,183,200,164,183,200,164
185           DFB   218,255,128,165,255,128,165
186           DFB   254,237,183,219,237,183,219
187           DFB   200,146,128,164,146,128,164
188           DFB   255,219,237,255,219,237,255
189           DFB   219,128,200,128,128,200,128
190           DFB   129,164,146,137,164,146,137
191           DFB   200,129,164,129,129,164,129
192           DFB   128,128,164,200,128,164,200
193 *
194 * VARIABILI
195 *
196 SALVIR    -     $AEFD
197 GETVAL    -     $B79E
198 CHKOUT    -     $FFC9
199 CHROUT    -     $FFD2
200 CLALL     -     $FFE7
201 OPEN      -     $FFC0
202 SETLFS    -     $FFBA
203 SETNAM    -     $FFBD
204 ILQUER    -     $B248
205 SA        -     $0338
206 SX        -     $0339
207 SY        -     $033A
208 R         -     $033B
209 C         -     $033C
210 CONT      -     $033D
```

UN MUSICANTE DI NOME C/16

Parlare di C/16 o Plus/4 è un po' come parlare di Amiga (e non è una battuta): si decantano spesso le capacità grafiche a discapito di altri aspetti ugualmente degni di nota

di **Roberto Ferro**

Sappiamo tutti che il Commodore 16 mette a disposizione diverse possibilità circa la gestione della pagina grafica, ma è altresì vero che all'interno del Ted Chip esistono alcuni registri dedicati alla generazione del suono a cui fanno riscontro le potenti e semplici istruzioni Basic SOUND e VOL che sono sovente bistrattate e considerate di poca utilità.

Per sfatare questo luogo comune, stavolta presentiamo un breve programma Basic che mette finalmente in luce le potenzialità dei piccoli computer in campo musicale.

Con il programma "Musicus" possiamo scrivere i nostri spartiti direttamente sul video del C/16, Plus/4 e C/128 servendoci del joystick (in porta 1) per selezionare una tra le 13 diverse opzioni previste: scrivere le note, cancellarle, cambiare pentagramma, inserire pause, ascoltare ciò che abbiamo scritto e altro ancora. Esaminiamolo in dettaglio.

SCHEDA TECNICA

Software per applicazioni musicali

Hardware richiesto: C/16 (espanso oppure no), Plus/4, C/128; non adattabile ad altri computer Commodore

Richiede joystick

Anche i programmi pubblicati in queste pagine sono contenuti nel disco "Directory" di questo mese.

SUONARE CON IL COMMODORE 128

Il programma pubblicato, in effetti, è scritto per il C/16 e non è stato possibile effettuare tutte le modifiche necessarie per adattarlo totalmente anche al C/128.

Il programma, tuttavia, funziona in modo sufficientemente corretto anche con questo computer (le schermate grafiche di queste pagine sono state ottenute proprio con il C/128) ma alcune righe Basic sono da sistemare opportunamente.

Notevole è la diversa resa sonora dovuta al fatto che i parametri del comando Sound (riga 78) generano frequenze musicali diverse a seconda del computer usato. Non dovrebbe esser difficile, però, modificare i valori delle note incluse nel listato.

Importante, a nostro parere, è la possibilità di utilizzare la parte grafica del programma proposto e di sfruttare la facilità di selezione delle note e delle opzioni. I più bravi potranno aggiungere altre icone e pentagrammi in modo da sfruttare meglio la pagina grafica. Anche il comando Play, decisamente più versatile nel C/128, può sostituire il più semplice Sound per generare suoni di tutto rispetto.

MUSICUS

La schermata del programma è in pagina grafica, e si presenta costituita da un pentagramma con annessa chiave di violino; in basso vi sono tante piccole icone che indicano le diverse possibilità di scelta a nostra disposizione. Un cursore triangolare, guidato dal joystick, permetterà di posizionarsi sotto una delle icone per selezionarla. Il joystick, in questo programma, viene usato in maniera molto paricolare: con il movimento sinistra - destra abbiamo il controllo sulle opzioni, mentre con il movimento alto - basso possiamo spostare un secondo cursore, chiamato cursore - penna che si muove sul pentagramma. Il funzionamento e l'uso è dunque piuttosto intuitivo: con un movimento sinistra - destra selezioniamo l'opzione; con un movimento alto - basso decidiamo su quale riga del pentagramma posizionare la nota. Vediamo ora le 13 opzioni disponibili:

NOTE

Sono rappresentate dalle prime cinque icone. Possiamo scegliere tra le note di diversa durata: semicrome, crome, semiminime, minime e semibrevi. Come è noto i valori di queste sono in rapporto di 1 a 2 partendo dall'ultima fino alla semibreve; si dispone, quindi, della semicroma (che vale 1/16), della croma (1/8), della semiminima (1/4), della minima (1/2) e della semibreve (1).

Una volta selezionata la nota possiamo posizionare la penna sulla riga o sullo spazio che interessa e premere fire per vedere la nota disegnarsi sul pentagramma e sentire il suono generato.

DIESIS

E' noto che la scala musicale comprende sette note (cinque toni e due semitoni), dal Do al Si. Esistono, però, anche delle alterazioni che elevano il suono di un semitono cromatico; tali alterazioni vengono indicate con i diesis: avremo un Do diesis, un Re diesis e così via. Per indicarle sul pentagramma dovremo prima selezionare l'icona con il simbolo del diesis (#), e poi la nota desiderata.

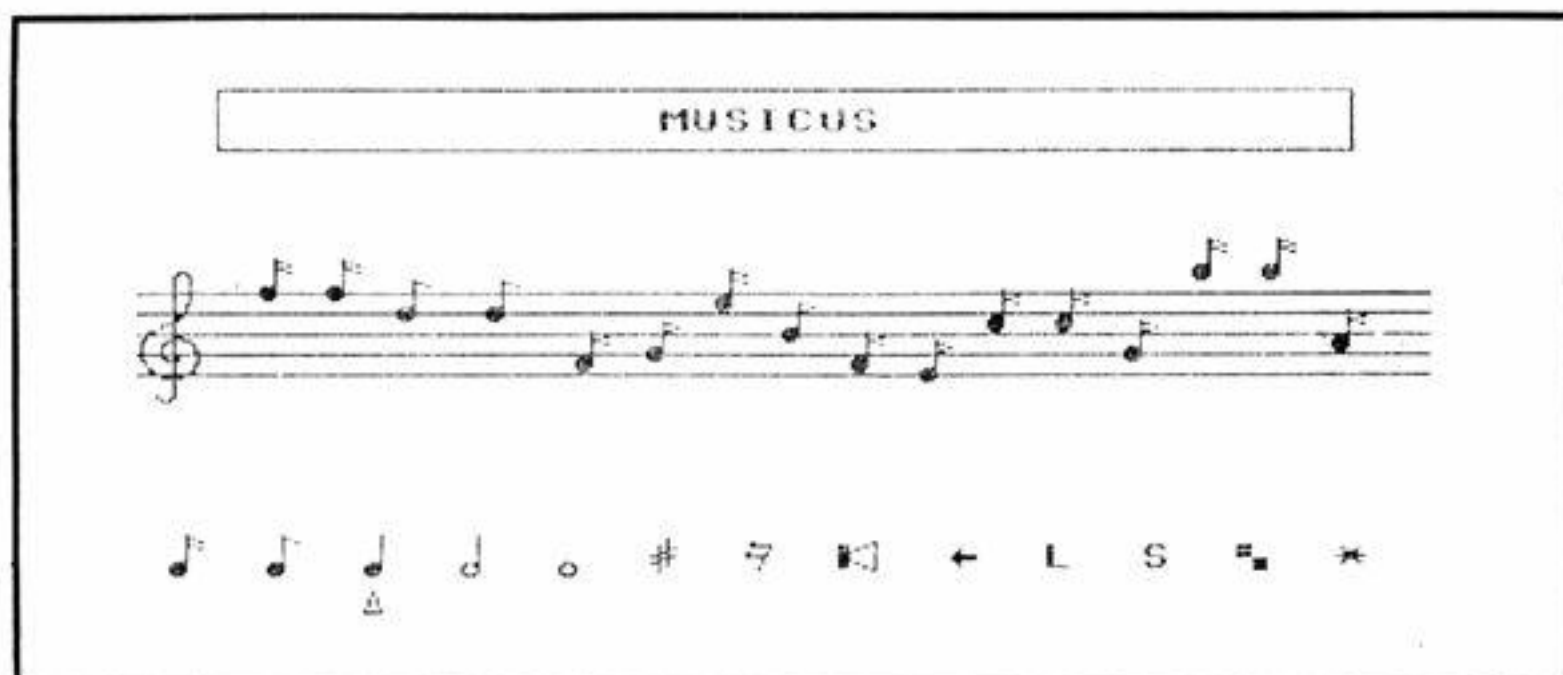
PAUSA

La disponibilità di pause è ridotta ad una sola, della durata di 1/16; si tratta, cioè, di una pausa di semicroma. La mancanza degli altri tipi di pause non è affato un fattore limitante in quanto per ottenere, ad esempio, una pausa di croma (1/8) sarà sufficiente indicare due pause di semicroma consecutive.

ASCOLTO

L'icona di ascolto è l'ottava, rappresentata da uno schematico altoparlante. Selezionandola potremo ascoltare l'intero brano composto fino a quel momento.

UNDO

Questa opzione, rappresentata dalla freccetta verso sinistra, serve a cancellare l'ultima nota, o pausa, nel caso avessimo compiuto un errore. In questo modo, nota per nota, è possibile ritornare indietro anche di tutte le note del brano per correggere ogni eventuale errore.

MUSICUS, IN BREVE

Diamo qui di seguito alcune note circa la struttura del programma per chi avesse intenzione di apportarvi modifiche:

0. Attiva i nuovi colori ed entra in pagina grafica;
2 - 26. Disegna la schermata contenente pentagramma ed icone;
28. Assegna le variabili per le posizioni dei cursori ed inizializza i vettori per le note (t% e l%)
30 - 52. Si occupa della gestione del joystick e del movimento dei due cursori.
54 - 90. Routine per la gestione delle varie opzioni di Ascolto, Note, Save, Load...;
92. Data contenenti le durate dei diversi tipi di note;
94 - 94. Data contenenti le frequenze delle note normali e diesis.
98. Routine di pausa;
100. Subroutine che preleva l'icona di una nota per stamparla successivamente sul pentagramma.

La struttura del programma, come si vede, non è complessa. Spesso si è cercato di racchiudere, in una stessa linea, un'intera funzione per i motivi di memoria di cui abbiamo già detto. Un accorgimento importante da tenere presente, nel caso di un Renumber, è quello di modificare convenientemente anche gli assegnamenti alla variabile W che sono presenti nele linee 28, 56 e 58.

Per aumentare il numero delle note bisogna invece agire sia sulla linea 28 (dove sono presenti le dichiarazioni di T% e L%), sia sulla linea 34, dove viene controllato che non si sia raggiunto il numero massimo i note immagazzinabili (K).

Ecco ora alcune variabili che possono tornare utili:

T% (n) :Contiene la frequenza della nota numero n;
L% (n) :Contiene la durata della nota n;
X, Y :Coordinate del cursore - penna;
C :Coordinata x del cursore - opzioni;
W :Puntatore alla linea contenente i DATA per le frequenze delle note;
K :Numero di note immagazzinate;
Z :Numero della periferica selezionata (1-8).

MEMORIZZA

Con questa opzione è possibile memorizzare il brano appena composto.

Selezionando la S(ave) apparirà la richiesta del nome del brano, poi dovremo scegliere la periferica tra 1 (nastro) o 8 (disk drive). Compiuta la registrazione avremo nuovamente a diposizione il nostro pannello di controllo con tanto di pentagramma.

CARICA

Permette di richiamare in memoria un brano precedentemente registrato. Anche in questo caso viene richiesto il suo nome ed il numero di periferica. A caricamento avvenuto possimo selezionare l'opzione Ascolto per sentire la musica, ma non sarà possibile visualizzare lo spartito sullo schermo in quanto, nel programma, non è implementata la routine di ri-visualizzazione del brano. E' importante fare molta attenzione alle operazioni di Memorizza e Carica in quanto le rispettive routine non sono dotate di nessun tipo di controllo o verifica sulla esistenza del file, la presenza e l'attività della periferica e così via: nel caso di errore il programma si interrompe con la perdita dei dati relativi al brano composto.

CAMBIO PAGINA

Giunti al termine di un rigo musicale, il programma si rifiuta di accettare altre note e, per segnalare l'impossibilità di proseguire, fa ascoltare il brano in memoria come se fosse stato selezionato Ascolto. Per accedere ad un nuovo rigo dobbiamo selezionare l'opzione Cambio Pagina che cancella il pentagramma correntemente visualizzato e ne presenta uno nuovo.

NUOVO BRANO

Questa opzione, che è l'ultima, è segnalata dall'asterisco. Selezio-

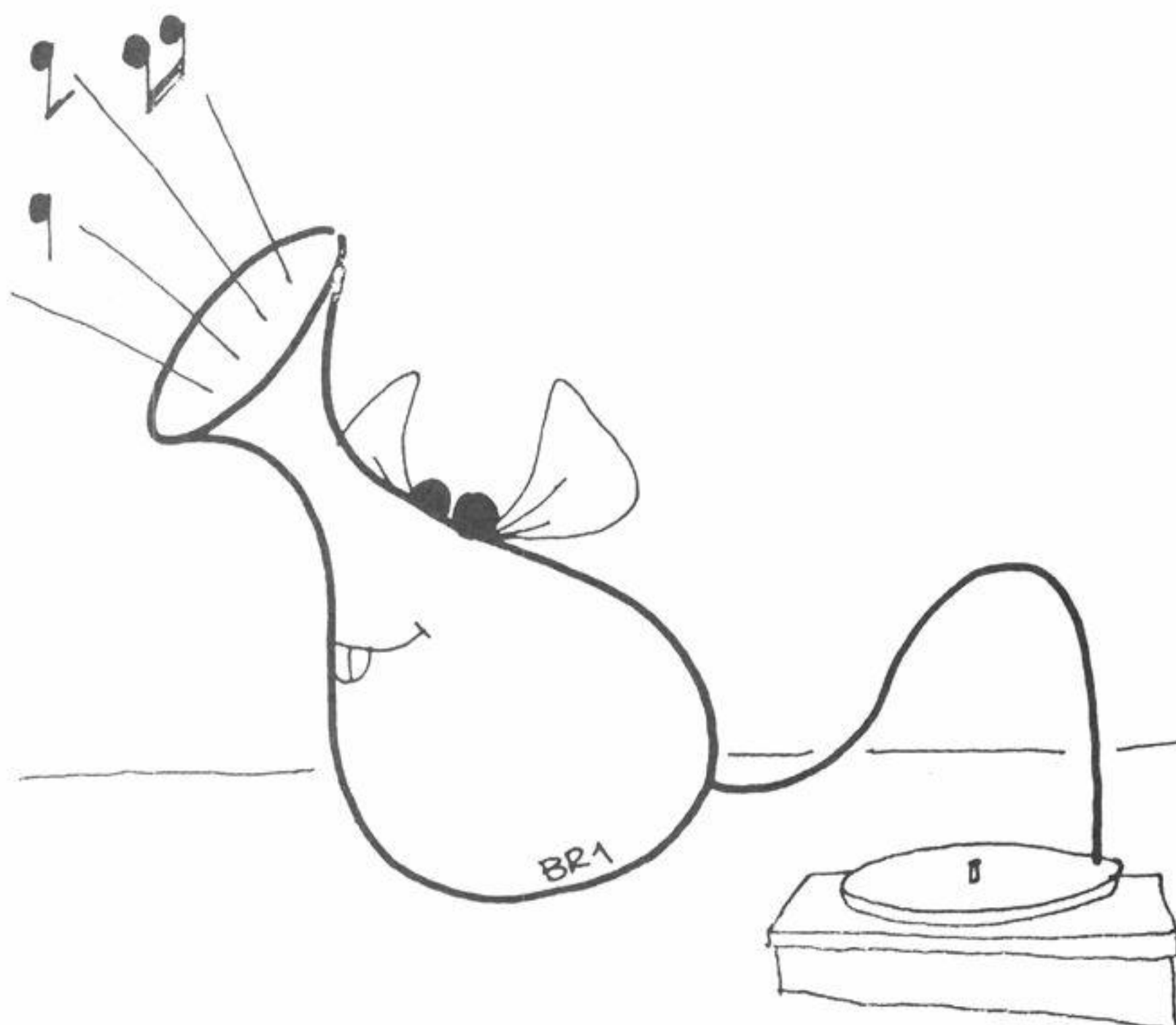
nandola abbiamo la possibilità di ricominciare la composizione exnovo, ma le note del brano in memoria vengono perse ed il pentagramma reinizializzato. Bisogna ovviamente usare con molta cautela questa scelta perchè non c'è un modo per recuperare i dati cancellati.

Dopo aver visto quali possibilità sono disponibili diciamo ancora che il limite massimo di note memorizzabili è fissato a 140 per motivi di memoria. E' chiaro che con un Plus/4 o un C/128 si può sensibilmente aumentare questo numero senza troppi problemi.

L'ESPANSIONE

La scarsità di memoria del piccolo C/16 ci costringe, anche stavolta, a ricorrere alla ormai celebre routine di espansione della memoria.

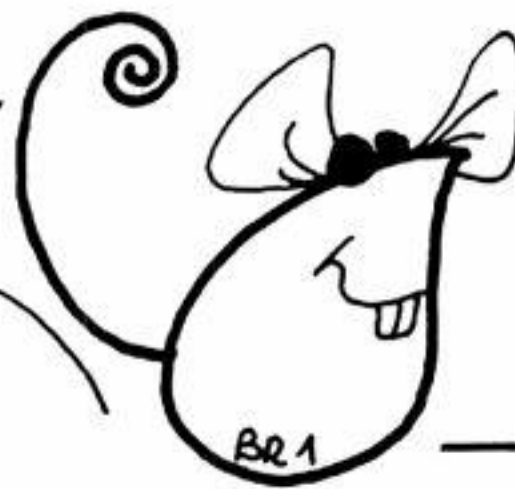
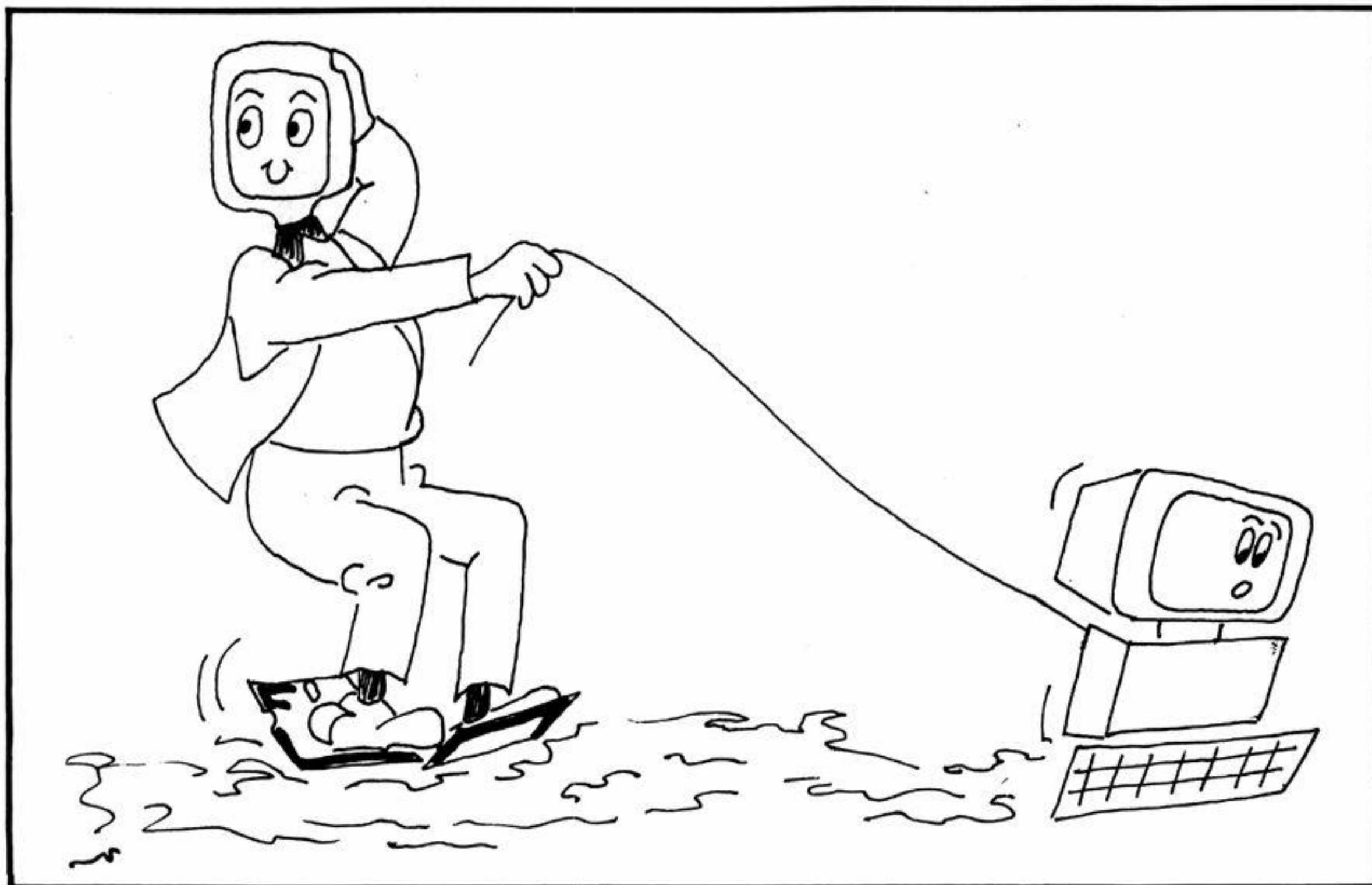
Diciamo subito che non è possibile utilizzare le routine proposte in passato, ma solamente quella di queste pagine, in quanto per MUSICUS la quantità di memoria richiesta è tale da dover sacrificare ancor maggiormente le dimensioni del video che arrivano a ridursi a due sole linee (80 caratteri!). La procedura per lanciare il programma, valida solo per il C/16 inespanso è, lo ricordiamo ancora una volta, la seguente:

- Digitare e salvare MUSICUS;
- Digitare e salvare ESPANSIONE MUSICUS;
- Lanciare ESPANSIONE MUSICUS attendendo alcuni secondi;
- Caricare e lanciare MUSICUS.

SUGGERIMENTI E MODIFCHE

Abbiamo già visto che sul C/16 il programma gira occupando TUTTA la memoria disponibile e pertanto è necessario che il programma venga ricopiato e salvato ESATTAMENTE come è pubblicato.

Qualcosa si può invece fare con il Plus/4. Un primo suggerimento può essere quello di irrobustire le routine di Memorizza e Carica per renderle più pronte ad eventuali errori; poi si potrebbe fare in modo che un brano registrato, e poi ricaricato, sia mostrato anche sul pentagramma. Oltre a questo si possono fare ulteriori modifiche più complesse, come ad esempio aggiungere la possibilità di gestire due voci, oppure di visualizzare contemporaneamente due pentagrammi (la pagina grafica è grande...).

Infine non rimane che un ultimo avvertimento circa le eventuali modifiche da apportare al programma: nel caso di un Renumber bisogna prestare moltissima attenzione alla linea 58 dove il Restore è usato con una variabile che viene inizializzata appositamente all'inizio del programma e alle linee 56 e 58 per puntare ai DATA relativi alle frequenze delle note normali e diesis.

```
100 REM ESPANSIONE V1.2  PER C/16 DA CARICARE E LANCIARE PRIMA DI
110 REM ATTIVARE IL PROGRAMMA "MUSICUS"
120 :
130 SYS 65409: POKE 2021,1
140 FOR I=2128 TO 3071: POKE I,1: POKE I+1024,0: NEXT
150 POKE43,81:POKE44,12:POKE45,83:POKE46,12:POKE47,83:POKE48,12:POKE49,83:POKE50
,12
160 END
```

```
0 COLOR4,2,0:COLOR0,2,0:COLOR1,6,5:GRAPHIC1,1:VOL8:CHAR,16,3,"MUSICUS"
2 COLOR1,9,3:BOX,20,18,300,36:COLOR1,1,5:FORI=10TO106STEP24:CIRCLE,I,159,2,2:NEX
T
4 FORI=12TO84STEP24:DRAW,I,149TOI,159:NEXT
6 FORI=10TO58STEP24:PAINT,I,159:NEXT:FORI=0TO24STEP24:DRAW,13+I,150TO16+I,151:NE
XT
8 DRAW,12,151TO16,153:DRAW,126,155TO132,153:DRAW,126,157TO132,155
10 DRAW,128,152TO128,160:DRAW,130,150TO130,158
12 BOX,151,151,152,152:BOX,150,154,151,155:DRAW,156,152TO153,159
14 DRAW,150,155TO154,155:DRAW,153,152TO156,152:CHAR,25,19,"←  L  S  ▚  *"
16 DRAW,175,154TO182,151TO182,159TO175,156TO175,154:BOX,173,152,175,158,,1
18 COLOR1,1,5:FORI=78TO102STEP6:DRAW,0,ITO320,I:NEXT
20 FORI=1TO6STEP5:CIRCLE,9,97,I,I,,180:NEXT
22 FORI=3TO8STEP5:CIRCLE,9,95,I,I,180:NEXT
24 CIRCLE,9,77,4,9,65,180
26 DRAW,9,75TO9,108:CIRCLE,7,108,2,2,90,270:CIRCLE,11,74,2,2,270,90
28 DIMT%(140),L%(140):C=10:Y=87:X=30:W=94:GOTO50
30 DRAW,X,Y:J=JOY(1):ONJGOTO38,38,40,40,42,42,44,44
32 IFJ=0THEN30
34 G=(C+14)/24:IFG<8AND(K>139ORX>310)THEN78
36 ONGGOTO54,54,54,54,54,56,60,78,80,64,64,88,90
38 B=-3:GOTO46
40 A=24:GOTO50
42 B=3:GOTO46
44 A=-24:GOTO50
46 IFY+B>110ORY+B<71THEN30
48 DRAWT,X,Y:Y=Y+B:LOCATEX,Y:T=RDOT(2):DRAW,X,Y:GOTO98
50 IFC+A<10ORC+A>298THEN30
52 CIRCLE0,C,170,2,4,,,,120:C=C+A:CIRCLE,C,170,2,4,,,,120:GOTO30
54 GOSUB100:GSHAPEA$,X-2,Y-10,2:D=(C+14)/24*5+12:X=X+D:D=0:GOTO58
56 W=96:GOSUB100:GSHAPEA$,X-8,Y-6,2:GOTO30
58 RESTOREW:W=94:FORI=1TO(Y-72)/3+1:READM:NEXT:GOTO62
60 M=1022:F=144:GOSUB100:GSHAPEA$,X,83,2:D=17:DRAWT,X,Y:X=X+D
62 RESTORE92:FORI=0TO(C-10-F)/24:READH:NEXT:F=0:GOTO76
64 GRAPHIC0:FORI=1319TO1328:POKEI,0:NEXT:INPUT"]NOME";N$:INPUT"1N/8D";Z:IFG=10TH
EN70
66 IFZ=1THEN68:ELSEN$=N$+",S,W"
68 OPEN1,Z,Z,N$:PRINT#1,K:FORI=0TOK:PRINT#1,T%(I):PRINT#1,L%(I):NEXT:CLOSE1:GOTO
74
70 IFZ=1THEN72:ELSEN$=N$+",S,R"
72 OPEN1,Z,0,N$:INPUT#1,K:P=0:DO:INPUT#1,T%(P),L%(P):P=P+1:LOOPUNTILP=K:CLOSE1
74 GRAPHIC1:GOTO30
76 L%(K)=H:T%(K)=M:SOUND1,T%(K),L%(K):K=K+1:GOTO98
78 FORI=0TOK-1:SOUND1,T%(I),L%(I):FORJ=1TOL%(I)*11:NEXTJ,I:GOTO30
80 DRAW0,X,Y:K=K-1:IFK<0THENK=0:GOTO30
82 D=12:J=L%(K):T%(K)=0:RESTORE92
84 DO:READM:D=D+5:LOOPUNTILM=J:IFX-D>29THENX=X-D:BOX0,X-8,62,X+8,113,,1
86 FORI=78TO102STEP6:DRAW,0,ITO320,I:NEXT:GOTO30
88 BOX0,20,59,319,113,,1:X=30:GOTO86
90 K=0:GOTO88
92 DATA7,14,28,56,112
94 DATA897,881,864,854,834,810,798,770,739,704,685,643,596
96 DATA904,889,872,864,844,822,810,784,754,721,704,664,619
98 FORI=0TO50:NEXT:GOTO30
100 SSHAPEA$,C-4,161,C+6,149:RETURN
```

UNA PROTEZIONE TEMPORIZZATA

Sono molto diffusi, soprattutto al di fuori dell'informatica, sistemi di protezione a tempo; vediamo come applicarne uno per proteggere un programma

di **Michele Maggi**

Di recente vengono messi in commercio apparecchi elettronici il cui funzionamento è subordinato all'inserimento di un particolare codice da parte dell'utente.

Questi apparecchi (generalmente autoradio e simili) possiedono, al loro interno, un microprocessore che, elaborando il codice immesso, lo confrontano con quello di accesso e provvedono, in caso di corrispondenza, a consentire il regolare utilizzo dell'apparecchio.

Nel caso in cui, invece, i codici non corrispondano, scatta il meccanismo di protezione: viene nuovamente richiesto il codice (ma mai per un numero di volte superiore a tre) e se, una volta esauriti i tentativi, dovesse ancora mancare la corrispondenza fra i codici, verrà bloccato l'accesso all'apparecchio per un certo numero di ore.

Al termine del periodo di "blocco" verrà nuovamente ripetuta la fase di richiesta codice e di eventuale nuovo blocco finchè non verrà inserito il codice d'accesso corretto.

L'utilità pratica del sistema è evidente: anche in caso di furto le possibilità di utilizzo abusivo dell'apparecchio sono scarsissime (quasi nulle) e ciò dovrebbe fungere da deterrente per eventuali malintenzionati.

Tutte le considerazioni valgono anche per il software e, se intelligentemente utilizzate, possono fornire buone idee per sistemi di protezione.

PROTEGGIAMO

Il programma proposto in queste pagine funziona, appunto, con il metodo di cui sopra: non appena partito, chiederà una password (per un certo numero di volte specificato nella variabile NT) e si regolerà in funzione della risposta più o meno esatta.

E' predisposto per funzionare come subroutine facente parte di un programma più ampio ed è personalizzabile in modo da adattarlo alle più svariate esigenze.

La prima operazione che verrà effettuata, ancora prima di chiedere la password, sarà qualla di esaminare la presenza di un certo valore su un particolare blocco del dischetto.

La mancanza del valore (o la presenza del valore = 0) determineranno l'immediata richiesta della password; la presenza di un valore positivo (compreso tra 1 e 9) determinerà un "blocco" temporaneo sia in funzione del numero trovato sia in funzione dei parametri specificati nelle linee 62220 - 62230 (variabili R0 ed R1).

Naturalmente più numerosi saranno i tentativi errati più sarà alto il numero memorizzato sul dischetto (fino ad un massimo di 9).

Nel caso di inserimento di password corretta verrà invece sbloccato tutto il meccanismo, riportando a zero il valore del blocco sul dischetto.

Non si tratta, chiaramente, di un sistema infallibile perchè ogni protezione può essere aggirata, pur se con difficoltà più o meno sormontabili, compatibilmente con l'abilità dell'hacker.

Tuttavia qualche consiglio può giovare a chi ha intenzione di utilizzare questo tipo di protezione.

1) Compilare il programma con uno dei tanti compilatori in circolazione (magari usando il Blitz con il relativo "antidecompilatore").
2) Codificare la password oppure inserire un sistema a password multiple.
3) Non far circolare i propri programmi...

CODIFICHIAMO LA PASSWORD

Il sistema a password è certamente molto comodo ma è purtroppo facilmente aggirabile.

Il punto debole consiste nella possibilità da parte dell'hacker di esaminare il programma direttamente su disco e se, come in questo caso, la password è contenuta in una stringa, risulta assai facile intervenire per modificarla.

Una soluzione può consistere nella codifica della password, ad esempio scindendo la stringa in varie parti come è riportato nel programmino di esempio.

Altri metodi di codifica sono stati esaminati in precedenza su C.C.C.

Vediamo ora come è strutturato il listato e quali sono le parti più interessanti.

Come già si è detto il programma va considerato come facente parte di un "Main" per cui la prima istruzione, a parte eventuali inibizioni di Run / Stop ecc., deve necessariamente essere un Gosub alla subroutine di protezione (vedi linea 20).

Le linee 62010 - 62130 hanno la sola funzione di visualizzare un massaggio di presentazione che può naturalmente essere modificato a piacimento.

Le linee 62150 - 62230 provvedono a settare i parametri per la protezione assegnando alla variabile PW$ la password, ad NT il numero dei tentativi possibili prima di far scattare il blocco e a R0 ed R1 i coefficienti determinanti il ritardo.

Essendo tutte variabili possono essere modificate facilmente e, in particolare, R0 ed R1 che consentono di determinare il ritardo "punitivo".

La parte di programma che ricerca l'eventuale errore presente su disco è contenuta nelle linee 62250 - 62420.

Consiste nell'apertura di un file random che testerà il settore 0 della traccia 1 per vedere se sono presenti eventuali "errori".

Se la verifica è positiva scatterà il meccanismo di blocco a tempo che genererà un ritardo in funzione del numero trovato nella traccia e dei parametri specificati in R0 ed R1.

In ogni caso il controllo passa successivamente alla routine 62440 - 62540 che contiene la richiesta della password e la relativa routine di controllo.

La routine 62560 - 62710 provvede alla scrittura sul settore del dischetto sia per la "punizione" (ingresso in linea 62610) sia per lo sblocco (ingresso in linea 62630).

Nel caso si cercasse di togliere il dischetto durante la scrittura verrà visualizzata la scritta "RESET!!!" ad indicare che i programma si è accorto di questa azione da parte dell'utente.

Naturalmente al posto di limitarsi a scrivere "RESET" si portà provvedere ad un reset reale.

Nulla di particolare, invece, circa la routine di input controllato, contenuta nelle linee 62730 - 62820 che provvede a visualizzare un asterisco (*) ad ogni carattere digitato.

E' intuitivo che la tecnica di protezione descritta in queste pagine può essere pericolosa nel caso in cui si decida di usarla su dischetti "pieni " di listati e/o dati in genere.

Il programma, infatti, gestisce una particolare locazione del dischetto (traccia 1, settore 0), indipendentemente dalla sua appartenenza, o meno, ad un file importante. Si consiglia, pertanto, di applicare la tecnica di protezione ad un dischetto contenente un solo programma e che, in ogni caso, non utilizzi mai la traccia 0, settore 1.

IL COMPILATORE, QUESTO SCONOSCIUTO

La maggior parte dei lettori conoscerà senz'altro il concetto di "programma compilatore" ma è opportuno parlarne per chi non ne sa molto o ha dubbi al riguardo.

Il Basic, come è noto, è un linguaggio interpretato; ciò significa che ogni volta che il Sistema Operativo incontra un'istruzione deve verificarne la sintassi con conseguente perdita di tempo.

Una volta controllata la sintassi si passa all'esecuzione vera e propria del comando e quindi alla prosecuzione del programma.

E' intuitivo che queste operazioni sono lunghe e laboriose e finiscono per rallentare notevolmente l'esecuzione del programma.

Un programma compilatore in sostanza effettua alcune delle operazioni suddette una sola volta producendo un codice "oggetto" che, non necessitando di controlli sintattici, risulta assai più veloce.

Oltre alla maggior velocità, il programma compilato presenta alcune caratteristiche che ne fanno un ottimo prodotto utilizzabile per sistemi di protezione.

Primo fra tutti l'indecifrabilità del programma oggetto; ogni versione compilata risulta assai complessa in quanto non sempre appare come sequenza di operazioni logiche ma molto spesso come agglomerato di istruzioni apparentemente senza senso.

In secondo luogo la maggior parte dei compilatori inibisce il Run/Stop impedendo così l'uscita dal programma.

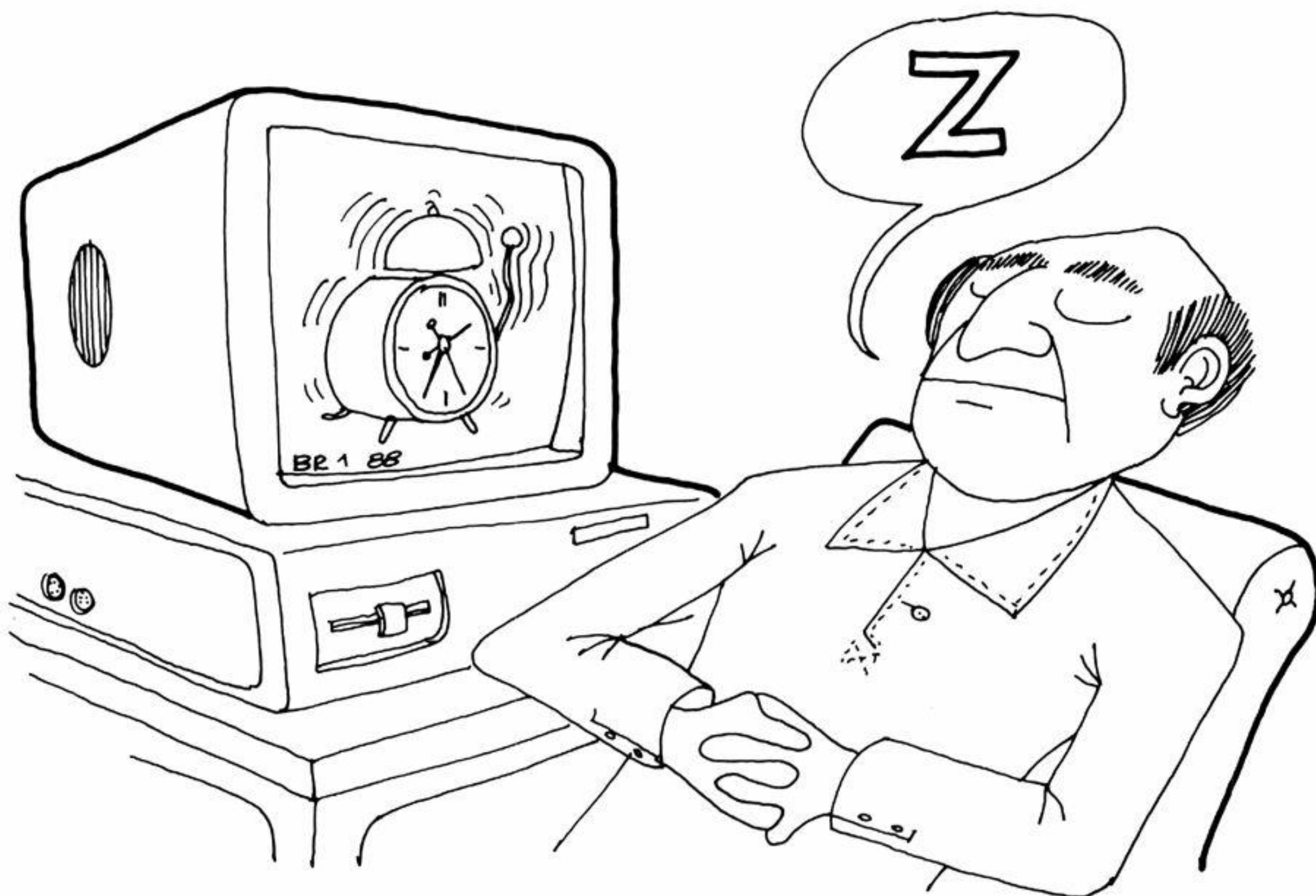
In terzo luogo i programmi compilati non sono listabili in quanto l'unica istruzione Basic è una SYS che passa il controllo al programma in Assembler.

Esistono in commercio vari compilatori le cui caratteristiche sono assai differenti.

I più diffusi sono Austrospeed, Petspeed, Abacus Compiler e Blitz.

Una esaustiva spiegazione delle loro differenze esula dallo spirito di questo articolo ma, qualora si manifestasse un certo interesse da parte dei lettori, sarà possibile realizzare un articolo ad hoc.

```
1 REM *---------------------*
2 REM * ESEMPIO DI POTEZIONE
3 REM * A TEMPORIZZATORE
4 REM * COMPLETAMENTE
5 REM * PERSONALIZZABILE
6 REM *---------------------*
7 REM * BY MICHELE MAGGI
8 REM * (C) SYSTEMS 1989
9 REM *---------------------*
10 :
11 REM   ATTENZIONE ALLA TRACCIA
12 REM   1, SETTORE 0 PERCHE' IL
13 REM    PROGRAMMA LA GESTISCE
14 REM PER APPLICARE LA PROTEZIONE.
15 REM  PER ATTIVARE IL PROGRAMMA
16 REM  DIGITA: RUN 20, DOPO AVER
17 REM    CANCELLATO LA RIGA 18
18 LIST 11-17
19 :
20 GOSUB 62060
30 PRINTCHR$(147)
40 FOR K=0 TO 999
50 POKE1024+K,INT(RND(1)*25)+1
60 POKE55296+K,INT(RND(1)*25)+1
70 NEXT
80 PRINTCHR$(147):END
90 :
62000 :
62010 REM *--------------------*
62020 REM * PRESENTAZIONE      *
62030 REM * DEL PROGRAMMA      *
62040 REM *--------------------*
62050 :
62060 PRINTCHR$(147);
62070 FOR I=1 TO 40
62080 PRINT CHR$(192);
62090 NEXT:PRINTCHR$(19);
62100 FOR I=1 TO 7
62110 PRINT CHR$(29);
62120 NEXT
62130 PRINT" SOFTWARE SECURITY SYSTEM "
62140 :
62150 REM *--------------------*
62160 REM * SETTAGGIO PARAMETRI*
62170 REM * PER L'ERRORE       *
62180 REM *--------------------*
62190 :
62200 PW$="MK64":REM PASSWORD
62210 NT=3  :REM NUMERO TENTATIVI
62220 R0=10 :REM MOLTIPLICATORE
62230 R1=30:REM COEFF. RITARDO
62240 :
62250 REM *--------------------*
62260 REM * RICERCA ERRORE     *
62270 REM * PREESISTENTE       *
62280 REM *--------------------*
62290 :
62300 OPEN 15,8,15
62310 OPEN 5,8,5,"#"
62320 PRINT#15,"B-R:5,0,1,0"
62330 INPUT#15,ER:IF ER <>0 THEN A=9:
      GOTO 62380
62340 PRINT#15,"B-P:5,6"
62350 :
62360 GET#5,A$:A=VAL(A$)
62370 IF A=0 THEN62420:REM NOERROR
62380 PRINT"ERRORE... ATTENDERE";A*R0
62390 FOR K=1 TO A*R0:PRINTK:PRINT"[illegible]":
      REM 2 CURSOR UP
62400 FORI=1TO R1:NEXTI
62410 NEXTK
62420 CLOSE5:CLOSE15
62430 :
62440 REM *--------------------*
62450 REM * RICHIESTA DELLA    *
62460 REM * PASSWORD D'ACCESSO *
62470 REM *--------------------*
62480 :
62490 FOR T=1 TO NT
62500 GOSUB62780
62510 IF PW$=PA$ THEN A=0:GOSUB62630:
      GOTO62540
62520 NEXT T
62530 GOSUB62610:GOTO 62300
62540 RETURN :REM ESEGUO IL PROGRAMMA
62550 :
62560 REM *--------------------*
62570 REM * GESTIONE SCRITTURA *
62580 REM * SU DISCHETTO       *
62590 REM *--------------------*
62600 :
62610 A=A+1
62620 IF A=10 THEN A=9
62630 OPEN 15,8,15
62640 OPEN 5,8,5,"#"
62650 PRINT#15,"B-P:5,5"
62660 PRINT#5,A
62670 PRINT#15,"B-W:5,0,1,0"
62680 INPUT#15,ER:IF ER= 0THEN62700
62690 PRINT"RESET!!!!":END:REM SYS64738
62700 CLOSE5:CLOSE15
62710 RETURN
62720 :
62730 REM *--------------------*
62740 REM * ROUTINE DI INPUT   *
62750 REM * CONTROLLATO        *
62760 REM *--------------------*
62770 :
62780 PRINT"INSERIRE CODICE: ";
62790 GET IK$:IF IK$=""THEN62790
62800 IF IK$=CHR$(13) THEN PRINT:RETURN
62810 PA$=PA$+IK$:PRINT"*";
62820 GOTO 62790
62821 END
```

```
100 REM ESEMPIO DI CODIFICA PASSWORD
110 :
120 A$="": REM ANNULLAMENTO STRINGA
130 FOR I=1 TO 10: READ A
140 A$=A$+CHR$(A-I):REM CODICE
150 NEXT: PRINT A$
160 :
170 DATA 66,78,72,87,88,71,85,76,91,89
```

ORE, OROLOGI E TEMPI VARI

Ecco come visualizzare permanentemente, sullo schermo del vostro C/64, un comodo orologio piuttosto preciso

di **Valerio Gionco**

Chi è quel tiranno che condiziona la nostra esistenza, dalla nascita alla morte, e che non si ferma mai?

E' il tempo, naturalmente.

Quello che si presenterà tra breve è un programma che "piazza" un orologio digitale nella prima riga dello schermo in modo da renderlo indelebile ad ogni tentativo di cancellazione (ad eccezione di Run / Stop e Restore). Il listato presenta, tra l'altro, un paio di aspetti particolarmente interessanti: il primo è che risiede (ovviamente) all'interno del ciclo di IRQ, e continua a girare indipendentemente da ciò che state facendo (entro certi limiti).

Il secondo, più importante, è che fa uso del clock chiamato TOD (Time Of Day) incorporato all'interno del CIA, o chip 6526.

Grazie a questo particolare, la routine l.m. è immune da qualunque tipo di rallentamento, dovuto, per esempio, ad operazioni di I/O e disabilitazioni dell'interrupt.

Ma vediamo in dettaglio il funzionamento dell'orologio, prima di passare all'esame del programma.

Innanzitutto precisiamo che nel C/64 esistono due circuiti integrati denominati CIA, che sovrintendono alla scansione della tastiera, alle operazioni di I/O ed altre cosucce. Potremmo, addirittura, azionare due orologi indipendenti contemporaneamente, ma per questa volta ci accontentiamo di uno solo.

I due chip 6526 sono così allocati in memoria:

CIA #1: $DC00 - $DCFF
CIA #2: $DD00 - $DDFF

Ed è proprio quest'ultimo (il CIA #27 che utilizzeremo. Le informazioni sono comun-

SCHEDA TECNICA

Software applicativo per tecniche di InterruptHardware richiesto: C/64, difficilmente adattabile ad altri computer CommodoreConsigliato agli esperti

Anche il programma pubblicato in queste pagine è contenuto nel disco "Directory" di questo mese.

Registro	Funzione
$08 (8)	TOD - decimi di secondo
$09 (9)	TOD - secondi
$0a (10)	TOD - minuti
$0b (11)	TOD - ore
$0d (13)	reg. contr. interrupt
$0f (15)	registro di controllo B

I registri del C.I.A. 6526

que valide per entrambi i chip.

Possiamo accedere all'orologio interno del CIA mediante i registri indicati in figura.

Possiamo in qualsiasi momento leggere il valore dei registri, ma dobbiamo ricordare alcune cose:

1) I valori dei secondi e dei minuti sono registrati in codice BCD, owero in modo decimale, TRANNE quelli delle ore.

2) Nel bit 7 del registro delle ore è presente il flag AM / PM: se esso è posto a 1 siamo in un'ora postmeridiana (PM) altrimenti in una antimeridiana (AM).

3) Quando leggiamo il registro delle ore, l' orologio SI BLOCCA TOTALMENTE, o meglio continua a funzionare solo internamente sinchè non leggiamo il registro dei decimi di secondo; subito dopo si aggiorna automaticamente e riparte.

Scrivere nei registri del CIA è leggermente più complesso che leggere in una Ram, a causa di una possibilità intrinseca del chip considerato: la possibilità di settare un allarme ad un'ora determinata.

Per far capire al 6526 se vogliamo scrivere una nuova ora nell'orologio o nell'allarme, usiamo il registro di controllo B, quello situato nel byte $0F.

Siamo più precisi: se il bit più significativo di tale registro è posto ad 1, stiamo settando l'allarme, altrimenti stiamo settando l'orologio.

Quando l'orologio del TOD raggiunge l'ora prefissata, il 6526 pone ad 1 il bit 2 del registro di controllo interrupt.

Anche qui vale la regola precedente: perchè l'orologio riparta dopo aver scritto nel registro delle ore, bisogna completare la scrittura dei decimi di secondo.

Tenete però presente che, in lettura, solo l'orologio è visibile, poichè i registri interni della sveglia sono a SOLA SCRITTURA: bisogna pertanto salvare, separatamente, i valori scritti in essi.

IL PROGRAMMA

Il programma è diviso in due parti: una è quella che si occupa dell'interpretazione del comando basic e della scrittura dell'ora nei registri del TOD, la seconda è quella che gira in interrupt e si occupa di leggere i registri del 6526 ed aggiornare continuamente l'orologio.

La sintassi da usare per l'attivazione dell' orologio è:

SYS 49152, 1, "hhmmssx"

in cui "hh" sono le ore (da 1 a 12); "mm" i minuti e "ss" i secondi.

Il carattere "x" sarà una "A" se si tratta di un'ora antimeridiana, ed una "P" se si tratta di un'ora postmeridiana.

Ad esempio...

SYS 49152, 1, "050610A"

...setta l'orologio alle ore 5:06 e 10 secondi del mattino.

Per disattivare l'orologio si usa la sintassi:
SYS 49152, 0

Naturalmente valori errati dei vari parametri causeranno la segnalazione di SYNTAX ERROR o ILLEGAL QUANTITY ERROR, a seconda dei casi.

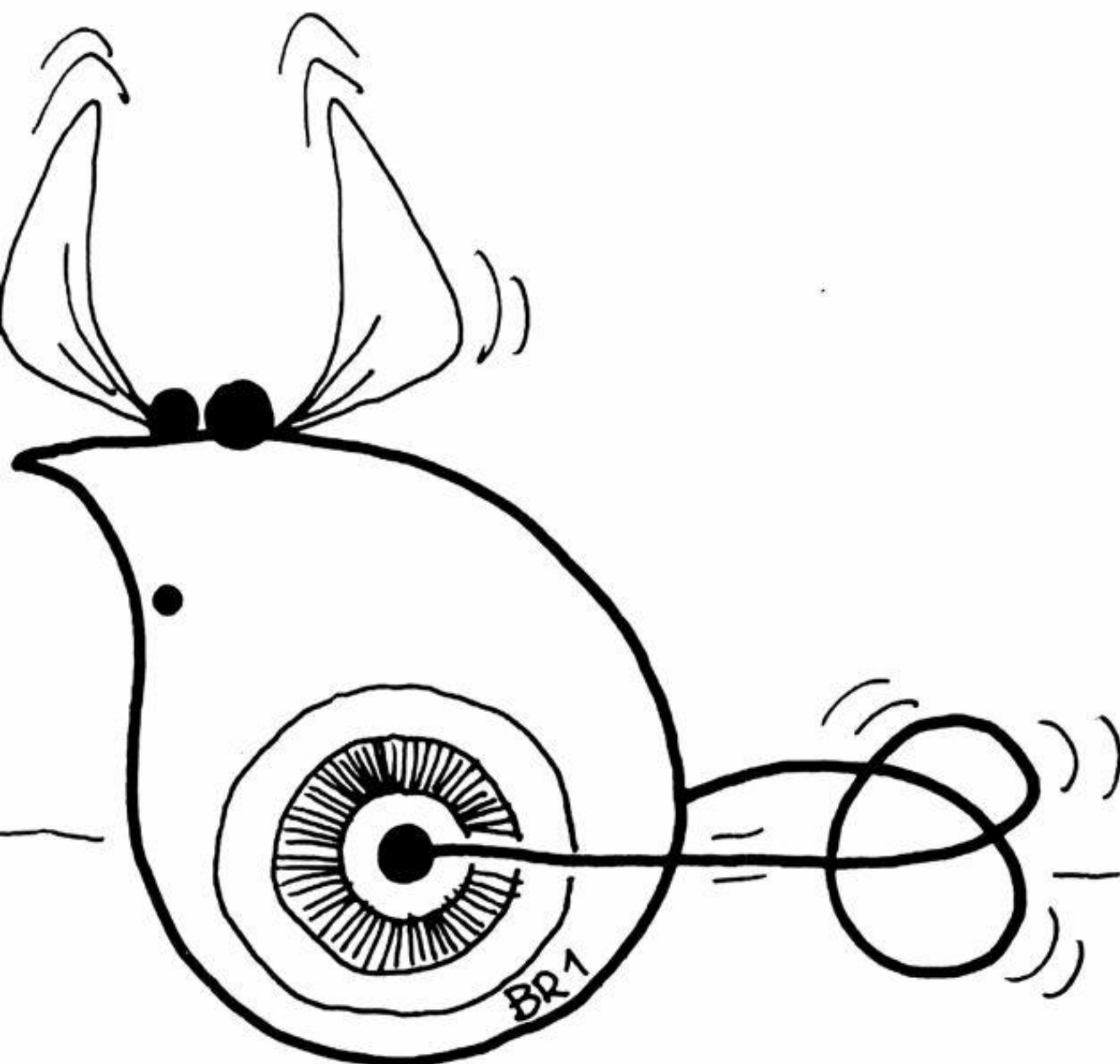
Premendo Run / Stop e Restore si disattiva egualmente la routine.

Per riportare, nell'orologio del CIA, lo stesso orario contenuto in TI$, sarà sufficiente...

Sys 49152, 1, TI$ + "A"

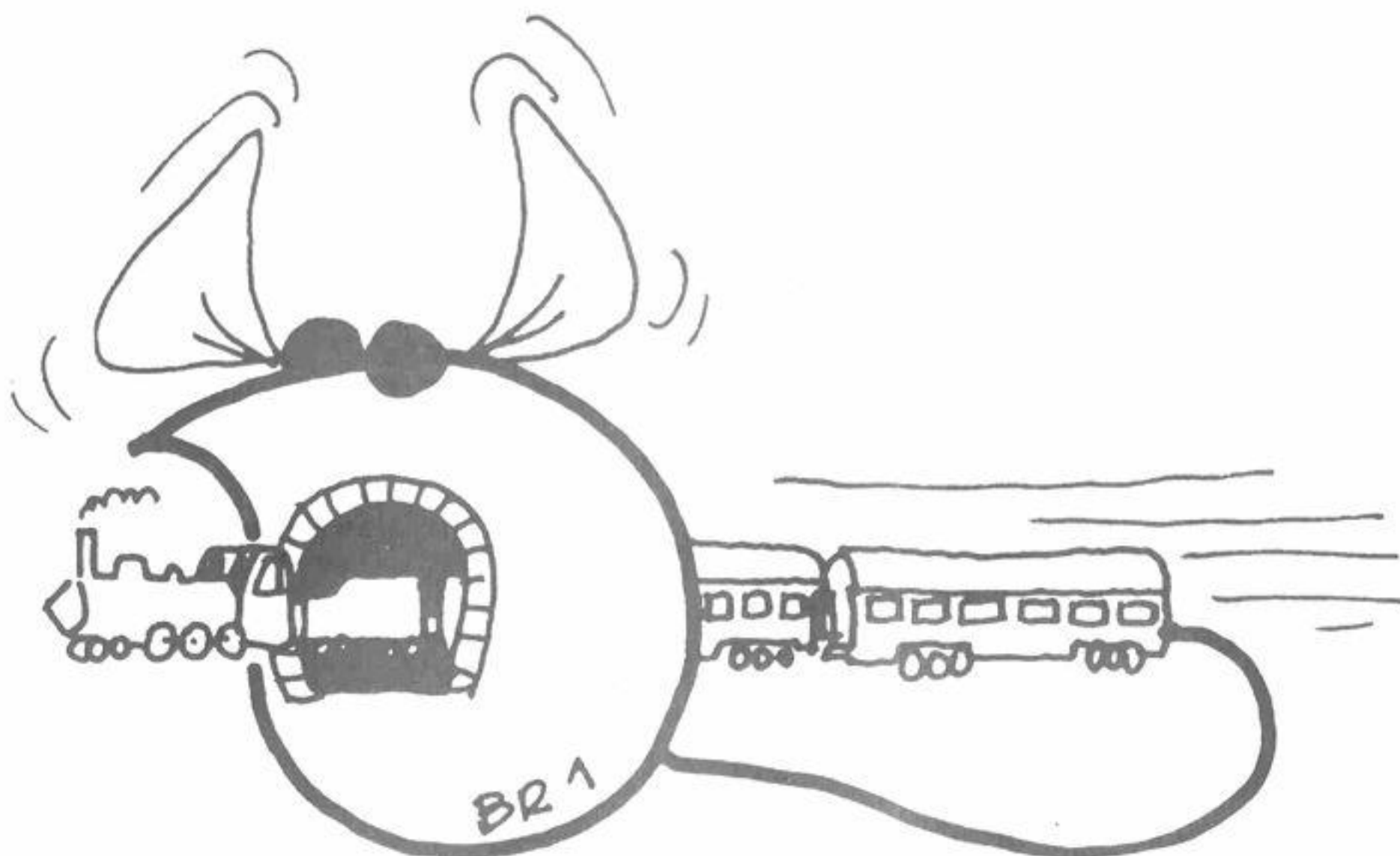
...oppure...

Sys 49152, 1, TI$ + "P"

...a seconda dei casi.

```
;
; T.O.D. clock v1.0 by Valerio Gionco
;
;     origine in $c000-49152
;


c000 nop            ;
     jsr $aefd      ;prende virgola
     jsr $b79e      ;prende numero
     cpx #$01       ;se>=1 setta l'ora
     bpl $c00e      ;se=0 cancella l'ora
     jmp $c17e      ;
c00e lda $dd0f      ;mette i registri
     and #$7f       ;del CIA in
     sta $dd0f      ;scrittura orologio
     jsr $aefd      ;prende una virgola
     jsr $ad9e      ;legge stringa
     jsr $b6a3      ;se la lunghezza
     cmp #$07       ;e' <>7 emette un
     beq $c026      ;syntax error
     jmp $af08      ;
c026 ldy #$00       ;carica il primo ca-
     lda ($22),y    ;rattere e controlla
     cmp #$30       ;che sia 0 o 1
     beq $c035      ;
     cmp #$31       ;
     beq $c044      ;
     jmp $af08      ;converte
c035 iny            ;
     lda ($22),y    ;
     jsr $c09c      ;
     sec            ;l'ora
     sbc #$30       ;
     sta $dd0b      ;
     jmp $c058      ;
c044 iny            ;
     lda ($22),y    ;in
     cmp #$30       ;
     bmi $c0a5      ;
     cmp #$33       ;
     bpl $c0a5      ;
     sec            ;esadecimale
     sbc #$30       ;
     clc            ;
     adc #$0a       ;
     sta $dd0b      ;e la mette in $dd0b
c058 jsr $c07d      ;prende i minuti
     sta $dd0a      ;e li scrive
     jsr $c07d      ;prende i secondi
     sta $dd09      ;e li scrive
     iny            ;
     lda ($22),y    ;prende un carattere
     cmp #$41       ;se e' una 'A' salta
     beq $c0ad      ;a c0ad
     cmp #$50       ;se non e' una 'P'
     bne $c07a      ;emette SYNTAX ERROR
     lda $dd0b      ;pone ad 1 il bit
     ora #$80       ;AM/PM
     sta $dd0b      ;
     jmp $c0ad      ;
```

```
c07a jmp $af08      ;emette syntax error
;
;subroutine che legge minuti e secondi
;
c07d iny            ;
     lda ($22),y    ;carica un carattere
     jsr $c09c      ;verifica sia valido
     sec            ;sottrae 30 dal suo
     sbc #$30       ;codice ASCII
     cmp #$06       ;se e' >=6 emette un
     bpl $c0a5      ;ILLEGAL QUANTITY
     asl a          ;
     asl a          ;
     asl a          ;moltiplica per 16
     asl a          ;
     sta $fb        ;lo mette in $fb
     iny            ;e preleva un' altro
     lda ($22),y    ;carattere
     jsr $c09c      ;controlla validita'
     sec            ;
     sbc #$30       ;sottrae 30
     ora $fb        ;aggiunge le decine
     rts            ;in $fb e ritorna
;
;subroutine controllo validita' caratt.
;
c09c cmp #$30       ;se il codice ascii
     bmi $c0a5      ;del carattere e'
     cmp #$3a       ;<30 o >39 emette un
     bpl $c0a5      ;ILLEGAL QUANTITY
     rts            ;
;
;subroutine ILLEGAL QUANTITY
;
c0ac jsr $c177      ;riavvia l'orologio
     ldx #$0e       ;carica in X l'erro-
     jmp ($0300)    ;re e lo emette


;
;inizializza l'orologio
;
c0ad jsr $c177      ;riavvia il TOD
     sei            ;
     lda #$bd       ;setta i vettori
     sta $0314      ;di interrupt per
     lda #$c0       ;puntare alla routine
     sta $0315      ;
     cli            ;
     rts            ;
;
;routine in interrupt
;
     dec $02        ;decrementa il flag
     bne $c0e6      ;del tempo: se e'=0
     lda #$0f       ;esegue la routine
     sta $02        ;
     jsr $c0e9      ;stampa l'ora
     jsr $c164      ;stampa i minuti
     lda #$20       ;scrive due spazi
     sta $0408      ;
     sta $0409      ;
```

```
     lda #$0d      ;scrive la 'M'
     sta $040b     ;
     lda #$3a      ;scrive i due punti
     sta $0402     ;fissi
     jsr $c125     ;stampa secondi
     jsr $c132     ;fa lampeggiare i :
     jsr $c13f     ;controlla AM/PM
c0e6 jmp $ea31     ;ed esce all'IRQ
;
;subroutine di stampa dell'ora
;
c0e9 lda $dd0b     ;legge registro ora
     and #$0f      ;prende i primi 4bit
     cmp #$0a      ;
     bpl $c0fd     ;
     adc #$30      ;
     sta $0401     ;converte
     lda #$30      ;
     sta $0400     ;
     rts           ;
     cmp #$0d      ;in
     bmi $c116     ;
     sec           ;
     sbc #$0c      ;
     sta $fb       ;
     lda $dd0b     ;ASCII
     and #$80      ;
     eor $fb       ;
     eor #$80      ;
     sta $dd0b     ;
     jsr $c177     ;
     rts           ;
c116 sec           ;e
     sbc #$0a      ;
     clc           ;
     adc #$30      ;
     sta $0401     ;
     lda #$31      ;
     sta $0400     ;stampa
     rts           ;
;
;legge e stampa i minuti

;
c164 lda $dd0a     ;legge registro
     jsr $c152     ;converte in ASCII
     sta $0403     ;e stampa
     sty $0404     ;
     rts           ;
;
;legge e stampa i secondi
;
c125 lda $dd09     ;legge registro
     jsr $c152     ;converte in ASCII
     sta $0406     ;e stampa
     sty $0407     ;
     rts           ;
;
;fa lampeggiare i duepunti (:)
;
c132 lda $dd08     ;legge i 1/10 di sec.
```

```
     cmp #$04      ;se >4 scrive i
     bmi $c14c     ;duepunti
     lda #$20      ;altrimenti li
     sta $0405     ;cancella
     rts           ;
c14c lda #$3a      ;
     sta $0405     ;
     rts           ;
;

;subroutine conversione BCD-ASCII
;
c152 tax           ;salva il val.in X
     and #$0f      ;prende i primi 4bit
     clc           ;
     adc #$30      ;aggiunge 30 e salva
     tay           ;in Y le unita'
     txa           ;riprende il valore
     and #$f0      ;prende i 4 bit piu'
     clc           ;significativi
     ror a         ;divide tutto per 16
     ror a         ;
     ror a         ;
     ror a         ;
     adc #$30      ;aggiunge 30
     rts           ;ed esce

;
;subroutine stampa AM/PM
;
c13f lda $dd0b     ;carica registro ore
     and #$80      ;prende bit n.7
     beq $c171     ;se =0 stampa una A
     lda #$10      ;altrimenti stampa
     sta $040a     ;una P
     rts           ;
c171 lda #$01      ;
     sta $040a     ;
     rts           ;
;
;riavvia l'orologio
;
c177 lda $dd08     ;legge e riscrive nel
     sta $dd08     ;registro dei decimi
     rts           ;riavviando il TOD
;

;spegne l'orologio
;
c17e sei           ;riporta l'indirizzo
     lda #$31      ;puntato da $314-315
     sta $0314     ;a $EA31
     lda #$ea      ;
     sta $0315     ;
     cli           ;
     lda #$20      ;e cancella i primi
     ldx #$0c      ;12 caratteri dello
c18e sta $03ff,x   ;schermo
     dex           ;
     bne $c18e     ;
     rts           ;
```

```
5 REM
6 REM   *** L'OROLOGIO T.O.D ***
7 REM ** BY VALERIO GIONCO 1988 **
8 REM
10 FOR X=0TO404:READ A
20 T=T+A:POKE49152+X,A:NEXT
30 READ C:IF T<>C THEN PRINT"ERRORE! RICONTROLLA I DATA!":STOP
32 :
33 SYS49152,1,"101234A"
40 DATA234,32,253,174,32,158,183,224,1
50 DATA 16,3,76,126,193,173,15,221,41,127,141,15,221,32,253,174,32,158
60 DATA 173,32,163,182,201,7,240,3,76
70 DATA 8,175,160,0,177,34,201,48,240,7,201,49,240,18,76,8,175,200,177
80 DATA 34,32,156,192,56,233,48,141,11,221,76,88,192,200,177,34,201
90 DATA48,48,90,201,51,16,86,56,233,48,24,105,10,141,11,221,32,125,192
100 DATA141,10,221,32,125,192,141,9,221,200,177,34,201,65,240,66
110 DATA201,80,208,11,173,11,221,9,128,141,11,221,76,173,192,76,8,175
120 DATA200,177,34,32,156,192,56,233,48,201,6,16,27,10,10,10,10,133
130 DATA251,200,177,34,32,156,192,56,233,48,5,251,96,201,48,48,5,201,58
140 DATA16,1,96,32,119,193,162,14,108,0,3,32,119,193,120,169,189
150 DATA 141,20,3,169,192,141,21,3,88,96,198,2
160 DATA208,37,169,15,133,2,32,233,192,32,100,193,169,32,141
170 DATA 8,4,141,9,4,169,13,141,11,4,169,58,141,2,4,32,37,193,32,50,193
180 DATA 32,63,193,76,49,234,173,11,221
190 DATA41,15,201,10,16,11,105,48,141,1,4,169,48,141,0,4,96,201,13,48
200 DATA 21,56,233,12,133,251,173,11,221,41,128,69
210 DATA 251,73,128,141,11,221,76,119,193,96,56,233,10,24,105,48
220 DATA 141,1,4,169,49,141,0,4,96,173,10,221,32,82,193,141,3,4,140
230 DATA 4,4,96,173,9,221,32,82,193,141
240 DATA 6,4,140,7,4,96,173,8,221,201,4,48,6,169,32,141,5,4,96,169,58
250 DATA 141,5,4,96,170,41,15,24,105,48,168,138,41,240,24,106,106,106
260 DATA106,105,48,96,173,11,221,41,128,240,6,169,16,141,10,4,96,169
270 DATA 1,141,10,4,96,173,8,221,141,8
280 DATA 221,96,120,169,49,141,20,3,169
290 DATA 234,141,21,3,88,169,32,162,12,157,255,3,202,208,250,96,41017
```

EMULATORE DI TURBO PASCAL PER C/64

E' finalmente pronto il potente package di emulazione prodotto dalla Systems Editoriale

Uno dei principali " difetti" del C/64 è paradossalmente rappresentato dal fatto che, per ciò che riguarda i linguaggi disponibili, questi tendono a sfruttare sino in fondo le caratteristiche della macchina.

Dal Simon's Basic di antica mèmoria, alle varie cartucce che implementano nuove, potenti istruzioni, numerose sono le novità e le comodità offerte all'utente del piccolo computer.

Purtroppo, però, linguaggi forse più " poveri", ma molto diffusi, non sono stati presi in considerazione dalle numerose software house che operano nel settore. Il risultato di tale " distrazione" consiste in un notevole ostacolo per l'utente del C/64 che intenda, gradualmente, passare verso sistemi più evoluti (e costosi).

Il Gw-Basic della Microsoft, ad esempio, è certamente povero per ciò che riguarda istruzioni sonore e non prende assolutamente in cosiderazione sprite ed altre amenità.

Come può arrangiarsi, pertanto, chi possiede il solo C/64 e segue, magari a scuola, un corso di informatica basato su Gw-Basic oppure sul Turbo Pascal? Tornato a casa, infatti, lo studente non può fare altro, per esercitarsi, che usare carta e penna e sperare che ciò che scrive sia accettato, l'indomani, dal suo insegnante.

La Systems Editoriale, invece, ha da tempo seguito la strada di offrire gli strumenti idonei per esercitarsi nei più popolari linguaggi usati nelle nostre scuole. Tempo fa la mossa si è dimostrata vincente: ben due edizioni di Ms-Dos emulator e Gw-Basic.

Oggi riproviamo con l'Emulatore Turbo Pascal per C/64.

COME E DOVE

Precisiamo subito che, a causa della struttura stessa del Pascal, è stata presa la decisione di offrire il software esclusivamente su disco, a prezzo decisamente popolare, quale deve esser quello di un programma per studenti.

E' ovvio che vi sono alcune (minime) differenze rispetto al T. Pascal originale Borland (cui si sono ispirati gli autori), tra cui, soprattutto, la velocità. Questa, come si può intuire, è strettamente legata alla rapidità tipica del trasferimento dei dati da/verso il drive 1541. Come nostra abitudine, precisiamo subito che il dischetto allegato alla confezione è privo di qualsiasi protezione: è possibile effettuare una copia del software servendosi di normalissimi copiatori.

Gli autori dell'Emulatore (gli ormai celeberrimi Mariani & Sorgato) hanno fatto di tutto affinchè l'utente del C/64 si trovasse ad operare come se avesse tra le mani un " vero" computer Ms-Dos compatibile: almeno nei casi meno complessi, infatti, un programma scritto in T. Pascal Borland (fatto girare su un PC) e lo stesso listato, attivato sul C/64 dotato di Emulatore, provocano lo stesso " effetto".

Inutile dire che, come nostra abitudine in casi similari, cureremo al più presto un'apposita rubrica T. Pascal in cui verranno divulgate le applicazioni più disparate.

ISTRUZIONI

Le istruzioni valide con l'Emulatore si suddividono in varie categorie. Vi sono quelle Standard (And, array, begin, case, const, div, do, downto, else, end, file, for, function, goto, if, in, label, mod, nil, not, of, or, packed, procedure, program, record, repeat, set, then, to, type, until, var, while, with). Gli identificatori sono: Abs, arctan, chr, cos, dispose, eof, eoln, exp, false, input, maxint, new, odd, ord, output, pred, read, readln, reset, rewrite, round, sin, sqr, sqrt, succ, text, true, trunc, write, writln. Oltre agli identificatori (boolean, char, integer, real, puntatore) vi sono: Goto, label, close, forwd, In, random, shr, shl, include.

GRAFICA E SUONO

Numerose sono le istruzioni che consentono di sfruttare lo schermo in alta risoluzione, tra cui ricordiamo le principali: Plot, draw, circle, fillscreen, fillshape, getdotcolor; anche Sound e Nosound sono presenti per consentire la massima compatibilità con il Turbo Pascal "originale".

Inutile dire che sono presenti altre istruzioni e che molte di queste rappresenteranno una piacevole sorpresa per chi già conosce il linguaggio compilatore.

L'altra sorpresa sarà sicuramente rappresentata dal prezzo di copertina: decisamente basso e, in ogni caso, alla portata di tutte le tasche.

PRINCIPIANTI, I DIECI COMANDAMENTI

1 Leggi attentamente il libretto di istruzioni del tuo computer e delle periferiche che ti sei procurato (registratore, drive, stampante, monitor, eccetera)

2 Non limitarti a leggere i listati di esempio ivi riportati, anche se sembrano banali: digitali sul computer e falli girare (digitando Run e premendo il tasto Return). Se non ti sono chiari, al contrario, la loro digitazione (e successiva esecuzione) ti chiarirà le idee.

3 Nel digitare i listati, ricordati di premere SEMPRE il tasto Return quando giungi alla fine del rigo, anche se tale operazione può sembrarti inutile.

4 Non confondere la vocale alfabetica "O" con il numero zero "0".

5 Digita sempre per esteso il comando PRINT e non abbreviarlo mai con il punto di domanda (?).

6 Inizia a digitare i listati più semplici e brevi: quelli più lunghi potrai digitarli quando avrai acquisito una maggior dimestichezza con il computer.

7 Dopo aver digitato un QUALSIASI programma, registralo subito, seguendo le istruzioni riportate sul manuale, PRIMA di dare Run.

8 Dopo aver fatto partire un programma, in caso, ad esempio, di segnalazione di errore in linea 350, digita soltanto...
List 350
...e accertati che la linea che appare sul video sia RIGOROSAMENTE identica a quella stampata sulla rivista.

9 Se, digitando un listato, ti accorgi che vi sono istruzioni di tipo Poke e Sys, raddoppia la prudenza nella digitazione e nelle procedure di registrazione PRIMA di dare Run.

10 Accertati che il programma che ti accingi a digitare sia REALMENTE valido per il tuo computer. Su ogni articolo della nostra rivista (da leggere SEMPRE con attenzione) è indicato il tipo di computer per il quale il listato stesso è idoneo.

A Scart - L. 28.000
Cavo di collegamento A500/A2000 con connettore per televisione SCART

Monitor a colori 1084 - L. 615.000
Monitor a colori ad alta risoluzione - Tubo 14" Black Matrix antiriflesso - Pitch 0.39 mm - Compatibile con Amiga 500/2000, PC (tutta la gamma), C64 e C128

Monitor a colori 2080 - L. 770.000
Monitor a colori ad alta risoluzione e lunga persistenza - Tubo 14" Black Matrix antiriflesso - Pitch 0.39 mm - Frequenza di raster 50 Hz - Compatibile con Amiga 500/2000, PC (tutta la gamma), C64 e C128

Monitor Monocromatico A2024 - L. 1.235.000
Monitor monocromatico a fosfori "bianco-carta" - Turbo 14" antiriflesso - (Disponibile da marzo '89)

PC60/40 - L. 8.930.000
Microprocessore Intel 80386 - Coprocessore matematico opzionale Intel 80387 - Clock 8 o 16 MHz selezionabile via software e da tastiera - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 funzioni - Sistema Operativo MS-DOS 3 2.1 - Interprete GW-Basic

PC60/40C - L. 9.180.000
Stessa configurazione ma con monitor 14" a colori mod. 1084

PC 60/80 - L. 10.450.000
Microprocessore Intel 80386 - Coprocessore opzionale Intel 80387 - Clock 8 o 16 MHz selezionabile via software e da tastiera - Memoria RAM: 2.5 MBytes - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Floppy Disk Drive opzionale da 3 1/2", 1.44 MBytes - 1 Hard Disk da 80 MBytes - 2 Porte parallele Centronics - Mouse video EGA (compatibile MDA - Hercules - CGA). Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Ambiente Operativo Microsoft Windows/386 - Interprete GW-Basic

PC60/80C - L. 10.700.000
Stessa configurazione ma con monitor 14" a colori mod. 1084

PC40/20 - L. 4.100.000
Microprocessore Intel 80286 - Coprocessore matematico opzionale Intel 80287 - Clock 6 o 10 MHz selezionabile via software, hardware o da tastiera - Memoria RAM: 1 MByte - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232 - Porta parallela Centronics - Scheda video AGA multistandard (MDA - Hercules - CGA) Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC40/20C - L. 4.350.000
Stessa configurazione ma con monitor 14" a colori mod. 1084

PC 40/40 - L. 5.285.000
Microprocessore Intel 80286 - Coprocessore matematico opzionale Intel 80287 - Clock 6 o 10 MHz selezionabile via software, hardware o da tastiera - Memoria RAM: 1 MByte - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232 - Porta parallela Centronics - Scheda video AGA multistandard (MDA - Hercules - CGA) Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC40/40C - L. 5.535.000
Stessa configurazione ma con monitor 14" a colori mod. 1084

1352 - L. 78.000
Mouse - Collegabile con Microsoft Bus Mouse - Collegabile direttamente a PC1, PC10/20 - III, PC40 - III

PC910 - L. 355.000
Floppy Disk Drive - Drive interno aggiuntivo da 3 1/2" per PC10/20-I-II-III - Capacità 360 o 720 KBytes selezionabile tramite "config. sys" - Corredo di telaio di supporto per l'installazione in un alloggiamento per un drive da 5 1/4" - Interfaccia identica ai modelli da 5 1/4"

PC1 - L. 995.000
Microprocessore Intel 8088 - 1 Floppy Disk Drive da 5 1/4" - Porta seriale RS232C - Porta parallela Centronics - - Monitor monocromatico 12" - Tastiera 84 tasti - Sistema Operativo MS-DOS 3.2 - Interprete GW-Basic

PCEXP1 - L. 640.000
PC Expansion Box - Box esterno di espansione per PC 1 - Alimentatore aggiuntivo incluso - Contiene 3 Slot di Espansione compatibili Ibm XT - Alloggiamento per Hard Disk da 5 1/4" - Si posiziona sotto il corpo del PC1 e viene collegato tramite degli appositi connettori

PC10-III - L. 1.965.000
Microprocessore Intel 8088 Clock 4.77 MHz 9.54 MHz (double) selezionabile via software e da tastiera - Memoria RAM: 640 KBytes - 2 Floppy Disk Drive da 5 1/4", 360 KBytes - Porta seriale RS232C - Porta parallela Centronics - Porta Mouse per Mouse Commodore 1352 (compatibile Microsoft Bus Mouse - Tastiera avanzata 102 con 12 tasti funzione Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC10-IIIC - L. 2.300.000
Stessa configurazione ma con monitor 14" a colori mod. 1084

PC20-III - L. 2.715.000
Microprocessore Intel 8088 - Clock 4.77 MHz 9.54 MHz (double) selezionabile via software e da tastiera - 1/4", 360 KBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232C - Porta parallela Centronics - Porta Mouse per Mouse Commodore 1352 (compatibile Microsoft Bus Mouse) - Tastiera avanzata 102 con 12 tasti funzione Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

QUANTO COSTA IL TUO COMMODORE

Amiga 2000 - L. 2.715.000

Microprocessore Motorola MC68000 - Clock 7.16MHz - Kickstart ROM - Memoria RAM: 1 MByte - 3 chip custom per DMA, Video, Audio, I/O - 5 Slot di Espansione Amiga Bus 100 pin Autoconfig™ - 1 Slot di Espansione 86 pin per Schede Coprocessore - 2 Slot di Espansione compatibili AT/XT - 2 Slot di Espansione compatibili XT - 2 Slot di Espansione Video - 1 Floppy Disk Drive da 3 1/2", 880 KBytes - Porta seriale RS232C - Sistema Operativo single-user, multitasking AmigaDOS - Compatibilità MS-DOS XT/AT disponibile con schede interne Janus (A2088 - A2286) - Monitor escluso

Amiga 500 - L. 995.000

Microprocessore Motorola MC68000 - Clock 7.16 MHz - Kickstart ROM - Memoria RAM: 512 KBytes - 3 Chip custom per DMA, Video, Audio, I/O - 1 Floppy Disk Driver da 3 1/2", 880 KBytes - Porta seriale RS232C - Porta parallela Centronics

Videomaster 2995 - L. 1.200.000

Desk Top Video - Sistema per elaborazini video semiprofessionale composto da genlock, digitalizzatore e alloggiamento per 3 drive A2010 - Ingressi videocomposito (2), RGB - Uscite Videocomposito, RF, RGB + sync -

Floppy Disk Driver A 1010 - L. 335.000

Floppy Disk Driver - Drive esterno da 3 1/2" - Capacità 880 KBytes - Collegabile a tutti i modelli della linea Amiga, alla scheda A2088 e al PC1

Floppy Disk Drive A 2010 - L. 280.000

Floppy Disk Drive - Drive interno aggiuntivo da 3 1/2" - Capacità 880 KBytes - Collegabile ad Amiga 2000

Hard Disk A 590 - L. 1.750.000

Hard Disk+Controller+RAM - Scheda Controller - Hard Disk da 3 1/2" 20 MBytes - 2 MBytes "fast" RAM - Collegabile all'Amiga 500

Scheda Janus A 2088 + A 2020 - L. 1.050.000

Scheda Janus XT+Floppy Disk Drive da 5 1/4", 360 KBytes - Scheda Bridgeboard per compatibilità MS-DOS (XT) in Amiga 2000 - Microprocessore Intel 8088 - Coprocessore matematico opzionale Intel 8087

A2286+A2020 - L. 1.985.000

Scheda Janus AT+Floppy Disk Drive da 5 1/4", 1.2 MBytes - Scheda Bridgeboard per compatibilità MS-DOS (AT) in Amiga 2000 - Microprocessore Intel 80287 - Clock 8 MHz - RAM: 1 MBytes on-board - Floppy Disk Controller on-board - Floppy Disk Driver disegnato per l'installazione all'interno dell'Amiga 2000 -

Scheda A2620 - L. 2.700.000

Scheda Processore Alternativo 32 bit - Scheda per 68020 e Unix - Microprocessore Motorola MC68020 - Coprocessore matematico Motorola MC68881 (opzionale MC68882)

Scheda A Unix - L. 3.250.000

Sistema Operativo AT&T Unix System V Release 3 - Per Amiga 2000 con scheda A2620 e Hard Disk 100 MBytes

Hard Disk A2092+PC5060 - L. 1.020.000

Hard Disk e controller - Hard Disk 3 1/2" ST506 - Capacità formattata 20 MBytes

Hard Disk A2090+2092 - L. 1.240.000

Hard Disk e controller - Hard Disk 3 1/2" ST506 - Capacità formattata 20 MBytes

Hard Disk A2090+A2094 - L. 1.900.000

Stesse caratteristiche del kit A2092 ma con disco da 40 MBytes

Espansione di memoria A2058 - L. 1.149.000

Espansione di memoria - Scheda di espansione per Amiga 2000 - Fornita con 2 MBytes "fast" RAM, espandibile a 4 o 8 MBytes

Scheda Video A2060 - L. 165.000

Modulatore video - Scheda modulatore video interna per Amiga 2000 - Uscite colore e monocromatica - Si inserisce nello slot video dell'Amiga 2000

Genlock Card A2301 - L. 420.000

Genlock - Scheda Genlock semiprofessionale per Amiga 2000 - Permette di miscelare immagini provenienti da una sorgente esterna con immagini provenienti dal computer

Professional Video Adapter Card A2351 - L. 1.500.000

Professional Video Adapter - Scheda Video Professionale per Amiga 2000 (B) - Genlock qualità Broadcast - Frame Grabber - Digitalizzatore - Include software di controllo per la gestione interattiva
(Disponibile da maggio '89)

A501 - L. 300.000

Espansione di memoria - Cartuccia di espansione di memoria da 512 KBytes per A500

A520 - L. 45.000

Modulatore RF - Modulatore esterno A500 - Permette di connettere qualsiasi televisore B/N o colori ad Amiga 500

PC20-IIIC - L. 3.050.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

Nuovo C64 - L. 325.000

Nuovo Personal Computer CPU 64 KBytes RAM - Vastissima biblioteca software disponibile - Porta seriale Commodore - Porta registratore per cassette - Porta parallela programmabile -

C128D - L. 895.000

Personal Computer CPU 128 KBytes RAM espandibile a 512 KBytes - ROM 48 KBytes - Basic 7.0 - Tastiera separata - Funzionante in modo 128,64 o CP/M 3.0 - Include floppy disk drive da 340 KBytes

Floppy Disk Drive 1541 II - L. 365.000

Floppy Disk Drive - Floppy Disk Drive da 5 1/4" singola faccia - Capacità 170 KBytes - Alimentazione separata - Compatibile con C64, C128, C128D

Floppy Disk Dirve 1581 - L. 420.000

Floppy Disk Drive da 3 1/2" doppia faccia - Capacità 800 KBytes - Alimentazione separata - Compatibile con C64, C128, C128D

1530 - L. 55.000

Registratore a cassette per C64, C128, C128D

Accessori per C64 - 128D

1700 - Espansione di memoria - Cartuccia di espansione di memoria a 128 KBytes per C128 **- L. 170.000**

1750 - Espansione di memoria - Cartuccia di espansione di memoria 512 KBytes per C128 - **L. 245.000**

1764 - Espansione di memoria - Cartuccia di espansione di memoria a 256 KBytes per C64
Fornita di alimentatore surdimensionato - **L. 198.000**

16499 - Adattatore Telematico Omologato - Collegabile al C64
Permette il collegamento a Videotel, P.G.E. e banche dati - **L. 149.000**

1399 - Joystick - Joystick a microswitch con autofire - **L. 29.000**

1351 - Mouse - Mouse per C64, C128, C128D - **L. 72.000**

Monitor Monocromatico 1402 - L. 280.000

Monitor monocromatico a fosfori "bianco-carta" - Turbo 12" antiriflesso - Ingresso TTL - Compatibile con tutta la gamma PC

Monitor Monocromatico 1404 - L. 365.000

Monitor monocromatico a fosfori ambra - Turbo 14" antiriflesso a schermo piatto - Ingresso TTL - Compatibile con tutta la gamma PC - Base orientabile

Monitor Monocromatico 1450 - L. 470.000

Monitor monocromatico BI-SYNC a fosfori "bianco-carta" - Turbo 14" antiriflesso - Ingresso analogico e digitale - Doppia frequenza di sincronismo orizzontale per compatibilità con adattatori video MDA, Hercules, CGA, EGA e VGA

Monitor a colori 1802 - L. 445.000

Monitor a colori - Turbo 14" - Collegabile a C64, C128, C128D

Monitor monocromatico 1900 - L. 199.000

Monitor monocromatico a fosfori verdi - Turbo 12" antiriflesso - Ingresso videocomposito - Compatibile con tutta la gamma Commodore

Monitor a colori 1950 - L. 1.280.000

Monitor a colori BI-SYNC alta risoluzione - Turbo 14" antiriflesso - Ingresso analogico e digitale - Doppia frequenza di sincronismo orizzontale per compatibilità con adattatori video MDA, Hercules, CGA, EGA e VGA

Stampante MPS 1230 - L. 465.000

Stampante a matrice di punti - Testina a 9 aghi - 120 cps - Bidirezionale - 80 colonne - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia seriale Commodore e parallela Centronics - Compatibile con tutti i prodotti Commodore

MPS 1230R - L. 19.000

Nastro per stampante

Stampante MPS 1500C - L. 550.000

Stampante a colori a matrice di punti - Testina a 9 aghi - 130 cps - Bidirezionale - 80 colonne - Supporta nastro a colori o nero - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia parallela Centronics - Compatibile con la gamma Amiga e PC

MPS1500R - L. 37.000

Nastro a colori per stampante

Stampante MPS 1550C - L. 575.000

Stampante a colori a matrice di punti - Testina a 9 aghi - 130 cps - Bidirezionale - 80 colonne - Supporta nastro a colori o nero - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia seriale Commodore e parallela Centronics - Compatibile con tutti i prodotti Commodore

I COMMODORE POINT

LOMBARDIA

Milano
• AL RISPARMIO - V.le Monza, 204
• BCS - Via Montegani, 11
• BRAHA ALBERTO
- Via Pier Capponi, 5
• E.D.S. - Corso Porta Ticinese, 4
• E.S.C. - Via Roggia Scagna, 7
• FAREF - Via A. Volta, 21
• FLOPPERIA - Viale Monte Nero, 31
• GBC - Via Cantoni, 7
- Via Petrella, 6
• GIGLIONI LAURA - Via D'Ovidio, 8
• GILGLIONI - Viale Luigi Sturzo, 45
• LOGITEK - Via Golgi, 60
• MARC[illegible] - Via F.lli Bronzetti, 37
• ME[illegible]IONI - Via P. Colletta, 37
• MESSAGGERIE MUSICALI
- Galleria del Corso, 2
• NEWEL - Via Mac Mahon, 75
• RIVOLA - Via Vitruvio, 43
Provincia di Milano
• IL CURSORE - Via Cavour, 35
- Novate Milanese
• REC ELGRA - Corso Milano, 118
- Bovisio Masciago
• F.LLI GALIMBERTI
- Via Nazionale dei Giovi, 28/36
- Barlassina
• GBC - Viale Matteotti, 66
- Cinisello Balsamo
• P. GIORGIO OSTELLARI
- Via Milano, 300 - Desio
• CASA DELLA MUSICA
- Via Indipendenza, 21
- Cologno Monzese
• PENATI - Via Verdi, 28/30
- Corbetta
• EMP - Viale Italia, 12 - Corsico
• CENTRO COMPUTER PANDOLFI
- Via Corridoni, 18 - Legnano
• COMPUTEAM - Via Vecellio, 41
- Lissone
• FUTURA - Via Solferino, 31 - Lodi
• M.B.M. - Corso Roma, 12 - Lodi
• L'AMICO DEL COMPUTER
- Viale Lombardia, 17
- Melegnano
• BIT 84 - Via Italia, 4 - Monza
• I.C.O. - Via dei Tigli, 14 - Opera
• NIWA HARD & SOFT
- Via Bruno Buozzi, 94
- Sesto S. Giovanni
• FERRARI LUIGI - Via Madre Cabrini
- Sant'Angelo Lodigiano
• IL COMPUTER SERVICE SHOP
- Via Padana Superiore, 197
- Vimodrone
Bergamo
• COMIF - Via Autolinee, 10
• CORDANI - Via dei Caniana, 8
• D.R.B. - Via Borgo Palazzo, 65
• NEW SYSTEMS - Via Paglia, 36
Provincia di Bergamo
• BERTULEZZI G. - Via Fantoni, 48
- Alzano Lombardo
• COMPUTER TEAM - Via Verdi, 1/B
- Carvico
• OTTICO ROVETTA
- Piazza Garibaldi, 6 - Lovere
• A.I.S. INTERNATIONAL
- Via San Carlo, 25
San Pellegrino Terme
• SISTHEMA - Via Roma, 45
- Sarnico
• COMPUTER POINT
- Via Lanfieri, 52 - Sarnico
• AB INFORMATICA
- Statale Cremasca, 66
- Urgnano
Brescia
• COMPUTER CENTER - Via Cipro, 6
• INFORMATICA 2000
- Via Stazione, 16/B
• MASTER INFORMATICA
- Via F.lli Ugoni, 10/B
• VIGASIO MARIO - Port. Zanardelli, 3
Provincia di Brescia
• MISTER BIT - Via Mazzini, 70 -Breno
• CAVALLI PIETRO
- Via 10 Giornate, 14/B
- Castrezzato
• VIETTI GIUSEPPE - Via Milano, 1/B
- Chiari
• MEGABYTE - Piazza Maluezzi, 14
- Desenzano del Garda
• CO-RE - Via XXV Aprile, 136 - Flero
• BARESI RINO & C.
- Via XX Settembre, 7 - Ghedi
• INFO CAM - Via Provinciale, 38
- Gratacasolo
Como
• IL COMPUTER - Via Indipendenza, 90
• 2M ELETTRONICA - Via Sacco, 3
Provincia di Como
• ELTRONGROS - Via L. da Vinci, 54
- Barzanò
• EGA - Via Mazzini, 42
- Cassago Brianza
• DATA FOUND - Via A. Volta, 4 -Erba
• EGA - Via A. Moro, 17 - Galbiate
• RIGHI ELETTRONICA
- Via G. Leopardi, 26
- Olgiate Comasco
• CIMA ELETTRONICA
- Via Leonardo da Vinci, 7
- Lecco
• FUMAGALLI - Via Cairoli, 48 -Lecco
Cremona
• MONDO COMPUTER
- Via Giuseppina, 11/B
• PRISMA - Via Buoso sa Dovara, 8
• TELCO - Piazza Marconi, 2/A
Provincia di Cremona
• ELCOM - Via IV Novembre, 56/58
- Crema
• EUROELETTRONICA
- Via XX Settembre, 92/A - Crema
Mantova
• TUBALDO ELSO & C.
- Galleria Fermi, 7
• 32 BIT - Via Cesare Battisti, 14
• ELETTRONICA BASSO
- V.le Risorgimento, 69
Pavia
• POLIWARE - C.so C. Alberto, 76
Provincia di Pavia
• LOGICA INFORMATICA
- Via Monte grappa, 32 - Vigevano
• M. VISENTIN - C.so V. Emanuele, 76
- Vigevano
Sondrio
• CIPOLLA M. - Via Tremogge, 25
Provincia di Sondrio
• FOTONOVA - Via Valeriana, 1
- San Pietro di Berbenno
Varese
• DIMECO SISTEMI - Via Garibaldi
• IL CENTRO ELETTRONICO
- Via Morazzone, 2
• SUPERGAMES - Via Carrobbio, 13
Provincia di Varese
• BUSTO BIT - Via Gavinana, 17
- Busto Arsizio
• CRESPI G.&C. - V.le Lombardia, 59
- Castellanza
• COMPUTER SHOP
- Via A. da Brescia, 2 - Gallarate
• LIMA IMPORT-EXPORT
- (Grandi Magazzini Bossi)
- Via Clerici, 196 - Gerenzano
• J.A.C. - Via Matteotti, 38
- Sesto Calende

VALLE D'AOSTA
Aosta
• F.lli GATTI - Via Festaz, 75

PIEMONTE
Alessandria
•BIT MICRO - Via Mazzini, 104
• SERVIZI INFORMATICI
- Via Alessandro III, 47
Provincia di Alessandria
• S.G.E. ELETTRONICA
- Via Bandello, 19 - Tortona
Asti
• RECORD - Corso Alfieri, 166/3
Cuneo
• ROSSI COMPUTER - C.so Nizza, 42
• STUDIO Software - C.so Nizza, 49
Provincia di Cuneo
• PUNTO BIT - Corso Langhe, 26/C
- Alba
• SDI - Via Vittorio Emanuele, 250
- Bra
• ASCHIERI GIANFRANCO
- Corso Emanuele Filiberto, 6
- Fossano
Novara
• ELCOM - Corso Mazzini, 11
• PROGRAMMA 3 - V.le Buonarroti, 8
• PUNTO VIDEO
- Corso Risorgimento, 39/B
Provincia di Novara
• MIRCO POLACCO & C.
- Via Monte Zeda, 4 - Arona
• ALL COMPUTER
- Corso Garibaldi, 106
- Borgomanero
• MICROLOGIC - Via Giovanni XIII, 2
- Domodossola
• ELLIOT COMPUTER
- Via Don Minzoni, 32 - Intra
Torino
• ALBA ELETTRONICA
- Via C. Fossati, 5/P
• ALEX COMPUTER
- Corso Francia, 333/4
• COMPUTER HOME
- Via San Donato, 46/D
• COMPUTING NEWS
- Via Marco Polo, 40/E
• DE BUG - C.so V. Emanuele II, 22
• DESME UNIVERSAL
- Via San Secondo, 95
• F.D.S. - Via Borgaro, 86/D
• INFORMATICA ITALIA
- Corso Re Umberto, 128
• MT INFORMATICA
- Corso Giulio Cesare, 58
• MUSICS'S Shop - C.so Potenza, 177
• NEW BUSINESS COMPUTER
- Via Nizza, 45/F
• PLAY GAMES - Via C. Alberto, 39/A
• RADIO TV MIRAFIORI
- C.so Unione Sovietica, 381
• SMIT ELETTRONICA
- Via Bibiana, 83/B
• TELERITZ - Corso Traiano, 34
Provincia di Torino
• PAUL E CHICO VIDEOSOUND
- Via V. Emanuele, 52 - Chieri
• BIT INFORMATICA
- Via V. Emanuele, 154 - Ciriè
• HI-FI CLUB - Corso Francia, 92/C
- Collegno
• I.C.S. - Stradale Torino N. 73
- Ivrea
• BAS - Corso Roma, 47
- Moncalieri
• CERUTTI MAURO
- Corso Torino, 234 - Pinerolo
• EUREX Corso Indipendenza, 5
- Rivarolo C.se
• DIAM INFORMATICA
- C.so Francia, 146/Bis - Rivoli
• FULLINFORMATICA
- Via Vittorio Veneto, 25 - Rivoli
• GAMMA COMPUTER
- Via Cavour, 3 A/B
- Settimo Torinese
Vercelli
• ELETTROGAMMA - C.so Bormida, 27
• ELETTRONICA di BELLAMO A. & C.
- Strada Torino, 15
Provincia di Vercelli
• C.S.I. TEOREMA - Via Losana, 9
- Biella
• SIGEST - Via Bertodano, 8 - Biella
• REMONDINO Franco - Via Roma, 5
- Borgosesia
• FOTOSTUDIO TREVISAN
- Via XXV Aprile, 24/B
- Cossato
• STUDIO FOTOGRAFICO IMARISIO
- Piazza Martiri Libertà, 7 - Trino

LIGURIA
Genova
ABM COMPUTER
- Piazza De Ferrari, 24/R
• CENTRO ELETTRONICA
- Via Chiaravagna, 10/R
- Località Sestri Ponente
• COMMERCIALE SOTTORIPA
- Via Sottoripa, 115/117
• FOTOMONDIAL
- Via del Campo, 3-5-9-11-13/R
• LA NASCENTE - Via San Luca, 4/1
• RAPPR - EL - Via Boreoratti, 23/R
Imperia
• CASTELLINO - Via Belgrano, 44
• SASA COMPUTER
- Via Nazionale, 256
Provincia di Imperia
• CENTRO HI-FI VIDEO
- Via della Repubblica, 38 - Sanremo
• CASTELLINO - Via Genova, 48
- Ventimiglia
La Spezia
• CATTONI - Via Vitt. Veneto, 75
• I.L. ELETTRONICA
- Via Vitt. Veneto, 123
Provincia di La Spezia
• I.L. ELETTRONICA - Via Aurelia, 299
- Fornola di Vezzano
Savona
• CASTELLINO
- Corso Tardy e Benech, 101
• ATHENA
- Via Carissimo E. Crotti, 16/R

VENETO
Belluno
• UP TO DATE
- Via Vittorio Veneto, 43
Provincia di Belluno
• GUERRA COMPUTERS
- Viale Mazzini, 10/C - Feltre

Padova
- BIT SHOP - Via Cairoli, 11
- COMPUMANIA
- Riviera Tiso Camposanpiero, 37
- COMPUTER POINT - Via Roma, 63
- D.P.R. - V.le Lombardo, 4
- GIANFRANCO MARCATO
- Via Madonna della Salute, 51/53
- ZELLA ADELIO
- Piazza De Gasperi, 31/A

Provincia di Padova
- CAERT - Via Andorra, 11
- Zona Industriale Camin

Treviso
- BIT 2000 - Via Brandolini d'Adda, 14

Provincia di Treviso
- DE MARIN - Vai XX Settembre, 74
- Conegliano
- SIDESTREET - Via S. d'Acquisto, 8
- Montebelluna
- FALCON - Via Terraggio, 116
- Preganziol

Venezia
- TELERADIO FUGA
- San Marco, 3457

Provincia di Venezia
- GUERRA EGIDIO & C.
- Via Bissuola, 20/A - Mestre
- TREKILOWATT
- Via Torre Belfredo, 47 - Mestre
- REBEL - Via F. Crispi, 10
- San Donà di Piave
- GUERRA COMPUTERS
- Via Vizzotto, 29
- San Donà di Piave
- TELFERT - Via Chiesa, 1509
- Sottomarina
- RADIOCESTARO - Via Roma, 89
- Spinea

Verona
- CASA DELLA RADIO - Via Cairoli, 10
- TELESAT - Via Vasco de Gama, 8

Provincia di Verona
- FERRARIN - Via dei Massari, 10
- Legnago

Vicenza
- ELETTRONICA BISELLO
- Viale Trieste, 427/429
- SCLACHI Market - Via Cà Balbi, 139

Provincia di Vicenza
- GUERRA COMPUTERS
- Via Dell'Industria - Alte Ceccato
- SCIAVOTTO - Via Zanella, 21
- Cavazzale
- COMPUTER B. COSTO
- Via del Costo, 34 - Thiene
- ELETTROCASA - Via Roma, 67
- Trissino

FRIULI VENEZIA GIULIA

Gorizia
- E.C.O. ELETTRONICA
- Via F.lli Cossar, 23

Pordenone
- RIGO - Viale Cossetti, 5

Provincia di Pordenone
- MDT - Piazza Repubblica, 5
- Villanova di Prata
- BRUNO DA PIEVE
- Via Colombera, 17 - Porcia

Trieste
- AVANZO GIACOMO
- Piazza Cavana, 7
- COMPUTER SHOP - Via P. Reti, 6
- COMPUTIGI - Via XX Settembre, 51
- CTI - Via Pascoli, 4

Udine
- MOFERT 2 - Via Leopardi, 21
- R.T. SYSTEM - Via L. da Vinci, 99

Provincia di Udine
- IDRENO MATTIUSSI & C.
- Via Liciniana, 50 - Tavagnacco

Rovigo
- CLINICA DEL RASOIO E DEL COMPUTER
- Via Fiume, 31/33

TRENTINO ALTO ADIGE

Bolzano
- C.M.B. ITALIA - Via Roma, 82
- MATTEUCCI PRESTIGE
- Via Museo, 54

Provincia di Bolzano
- ELECTRO TAPPEINER
- P.za Principale, 90 - Silandro
- RADIO MAIR - Via Centrale, 70
- Brunico
- ELECTRO RADIO HENDRICH
- Via delle Corse, 106 - Merano

Trento
- CRONST - Via Galilei, 25

EMILIA ROMAGNA

Piacenza
- COMPUTER Line - Via G. Carducci, 4
- DELTA COMPUTER
- Via Martiri della Resistenza, 15/G
- SOVER - Via IV Novembre, 60

TOSCANA

Arezzo
- DELTA SYSTEM - Via Piave, 13

Firenze
- ATEMA
- Via Bendetto Marcello, 1a/1b
- COOPERATIVA L.D.T. - Via Icaro, 9
- ELETTRONICA CENTOSTELLE
- Via Centostelle, 5/a
- HELP COMPUTER
- Via degli Artisti, 15/A
- PUNTO SOFT - Via Vagnetti, 17
- TELEINFORMATICA TOSCANA
- Via Bronzino, 36

Provincia di Firenze
- WAR GAMES
- Via Raffaello Sanzio, 126/A
- Empoli
- NEW E.V.M. COMPUTER
- Via degli Innocenti, 2
- Figline Valdarno
- COSCI F.LLI - Via Roma, 26 - Prato
- CENTRO INFORMATICA
- Via F.lli Cervi, 21/29 - Pontassieve

Grosseto
- COMPUTER SERVICE
- Piazza Ponchielli, 2

Livorno
- ETA BETA - Via San Francesco, 30
- FUTURA 2 - Via Cambini, 19

Provincia di Livorno
- ELETTRONICA ALESSI PAOLO
- Via Cimarosa, 1 - Piombino

Provincia di Lucca
- IL COMPUTER - V.le Colombo, 216
- Lido di Camaiore
- SANTI VITTORIO - Via Roma, 23
- San Romano Garfagnana

Massa
- EURO COMPUTER
- Piazza Bertagnini, 4
- FIRMWARE - Via Aurelia Ovest, 27

Provincia di Massa
- RADIO LUCONI - Via Roma, 24/B
- Carrara

Pisa
- C.H.S. - Via Carlo Cattaneo, 90/92
- ELECTRONIC SERVICE
- Via della Vecchia Tranvia, 10
- IT - LAB - Via Marche 8A/8B

Pistoia
- ELECTRONIC SHOP
- Via della Madonna, 49
- OFFICE DATA SERVICE
- Galleria Nazionale, 22

Provincia di Pistoia
- ZANNI & C. - Corso Roma, 45
- Montecatini Terme

Siena
- VIDEO MOVIE - Via Garibaldi, 17

Provincia di Siena
- ELECTRONIC Shop - Via A. Casini, 51
- Chianciano Terme
- ELETTRONICA
- Via di Gracciano nel Corso, 111
- Montepulciano

UMBRIA

Provincia di Perugia
- COMPUTER STUDIOS
- Via IV Novembre, 18/A
- Bastia Umbra

CAMPANIA

Provincia di Avellino
- FLIP FLOP - Via Appia, 68 -Atripalda

Benevento
- E.CO: INFORMATICA
- Via Pepicelli, 21/25

Caserta
- O.P.C. - Via G.M. Bosco, 24

Provincia di Caserta
- M.P. COMPUTER - Via Napoli, 30
- Maddaloni

Napoli
- BABY TOYS
- Via Cisterna dell'Olio, 5/Bis
- CASA MUSICALE RUGGIERO
- Piazza Garibaldi, 74
- CENTRO ELETTRONICO CAMPANO
- Via Epomeo, 121
- C.I.AN - Galleria Vanvitelli, 32
- DARVIN - Calata San Marco, 26
- ELETTRONICA RO.DA.LO.
- Via Epomeo, 216/B
- GIANCAR 2 - Piazza Garibaldi, 37
- GRUPPO BUSH
- Galleria Umberto I, 55
- ODORINO - Largo Lala, 22/A-B
- R 2 - Via F. Cilea, 285
- SPY - Via Fontana, 135
- TOP - Via S. Anna dei Lombardi, 12
- VIDEOFOTOMARKET
- Via S. Brigida, 19

Provincia di Napoli
- SPADARO - Via Romani, 93
- S. Anastasia
- TUFANO - S.S. Sannitica, 87 Km 7
- Casoria
- ELETTRONICA 2000
- Corso Durante, 40
- Frattamaggiore
- GATEWAY - Via Napoli, 68
- Mugnano
- NUOVA INFORMATICA SHOP
- Via Libertà, 185/191 - Portici
- BASIC COMPUTER
- C.so Garibaldi, 34
- Pozzuoli
- FALCO ELETTRONICA
- Via Sarno, 100 - Striano
- TECNOTRE - Via P. Fusco, 1/F
- Torre Annunziata

Salerno
- COMPUTER MARKET
- C.so Vitt. Emanuele, 23

Provincia di Salerno
- KING COMPUTER - Via Olevano, 56
- Battipaglia
- DIMER POINT - Via C. Rosselli, 20
- Eboli

PUGLIA

Bari
- ARTEL - Via G. d'Orso, 9
- COMPUTER'S ARTS
- Viale Meucci, 12/B

Provincia di Bari
- F. FAGGELLA - Corso Garibaldi, 15
- Barletta
- G. FAGGELLA
- Via P. d'Aragona, 62/A - Barletta
- G. LONUZZO - Via Nizza, 21
- Castellana

Brindisi
- MARANGI & MICCOLI
- Via Prov. San Vito, 165

Provincia di Foggia
- IL DISCOBOLO - Via T. Solis, 15
- San Severo

Lecce
- BIT
- Via 95° Regg.to Fanteria, 87/89

Provincia di Lecce
- CEDOK INFORMATICA - Via Roma, 31
- Tricase

Matera
- GUADIANO ELECTRONICS
- Via Roma, 1

Taranto
- ELETTROJOLLY - Via de Cesare, 13
- TEA - Via Regina Elena, 101

CALABRIA

Catanzaro
- C. & G. COMPUTER - Via F. Acri, 28
- PAONE SAVERIO - Via F. Acri, 93/99

Provincia di Catanzaro
- COMPUTER HOUSE - Via Bologna
- Crotone
- OTTICA FOTO NELLO RUELLO
- C.so Vittorio Emanuele, 177
- Vibo Valentia

Cosenza
- SIRANGELO COMPUTER
- Via N. Parisio, 25

Provincia di Cosenza
- ELIGIO ANNICHIARICO & C.
- Via Roma, 21 - Castrovillari
- ALFA COMPUTER
- Via Nazionale, 341/A
- Corigliano Scalo
- ING. FUSTO SALVATORE
- Corso Nicotera, 99
- Lamezia Terme

Reggio Calabria
- CONTROL SYSTEM
- Via S. Francesco da Paola, 49/DE
- SYSTEM HOUSE
- Via Fiume ang. Palestino, 1

Provincia di Reggio Calabria
- COMPUTER SHOP
- Via Matteotti, 50/52 - Locri

PRODOTTI SYSTEMS EDITORIALE

Software su cassetta

La voce III	L.12000
Raffaello	L.10000
Oroscopo	L.12000
Computer-Music	L.12000
Gestione familiare	L.12000
Banca dati	L.12000
Dichiarazione dei redditi (740/S)	L.16000
Matematica finanziaria	L.20000
Analisi di bilancio	L.20000
Arredare (richiede linguaggio Simon's Basic)	L.10000

Software su disco

Ms-Dos & Gw-Basic	L.25000
La voce III	L.12000
Oroscopo	L.12000
Gestione familiare	L.12000
Banca dati	L.12000
Dichiarazione dei redditi (740/S)	L.24000
Matematica finanziaria	L.20000
Analisi di bilancio	L.20000
Arredare (richiede linguaggio Simon's Basic)	L.20000
Graphic Expander C/128 in modo 80 colonne	L.27000

Offerta "Commodore speciale L.M." + dischetto L.16000

Libri

64 programmi per il Commodore 64	L. 4800
I miei amici C/16 e Plus/4	L. 7000
Strategie vincenti per Commodore 64	L. 5800
62 programmi per Vic 20, C/16 e Plus/4	L. 6500
Tutti i segreti dello Spectrum	L. 7000
Impara giocando il Basic dello Spectrum	L. 7000
Micro Pascal per Commodore 64/128	L. 7000
Dal registratore al drive del C/64	L. 7000
Ada	L. 5000
Il linguaggio Pascal	L. 5000

Directory

Ciascun dischetto	L.12000

Arretrati

Ciascun numero arretrato di Commodore Computer Club	L. 5000
Ciascun numero arretrato di Personal Computer	L. 5000
Ciascun numero arretrato di VR Videoregistrare	L. 5000

Sconti e agevolazioni

Le spese di imballo e spedizione sono a carico della Systems se ciascun ordine è pari ad almeno L.50000 (di listino).

Gli abbonati hanno diritto allo sconto del 10% e alla spedizione gratuita se la somma totale raggiunge la cifra di L.50000 (di listino).

Abbonamenti

Commodore Computer Club (11 fascicoli)	L. 50.000
Personal Computer (11 fascicoli)	L. 45.000
Commodore Computer Club + Personal Computer (11 + 11 fascicoli)	L. 95.000
VR Videoregistrare (12 numeri)	L. 50.000

N.B.: la cifra per gli abbonamenti non può essere conteggiata per ottenere gli sconti e le agevolazioni di cui sopra.

Compilate un normale modulo di C/C postale indirizzando a:

C/C postale N. 37952207
Systems Editoriale
Viale Famagosta, 75
20142 Milano

Non dimenticate di indicare chiaramente, sul retro del modulo (nello spazio indicato con "Causale del versamento") non solo il vostro nominativo completo di recapito telefonico, ma anche il materiale desiderato.

In ogni caso sarebbe opportuno inviare la presente scheda, debitamente compilata, allegando la fotocopia della ricevuta del versamento effettuato.

Chi volesse ricevere più celermente la confezione deve inviare la somma richiesta mediante assegno circolare oppure normale assegno bancario (non trasferibile o barrato due volte) intestato a: Systems Editoriale - Milano.

Per un ottimale utilizzo del software "Linguaggio Macchina e Routine grafiche per C/64" è opportuna la lettura del fascicolo "Commodore Speciale" appositamente dedicato.

Coloro che desiderano procurarsi i prodotti della Systems Editoriale devono inviare, oltre alla cifra risultante dalla somma dei singoli prodotti, la cifra di L.3000 per spese di imballo e spedizione, oppure L.6000 se si preferisce la spedizione per mezzo raccomandata.